ORSAY

N° d'ordre : 3264

THESE

PRESENTEE

A L'UNIVERSITE DE PARIS-SUD

CENTRE D'ORSAY

POUR OBTENIR

LE TITRE DE DOCTEUR 3ème CYCLE

Spécialité : ANALYSE NUMERIQUE

par

Agnès WIERNIK - MALGOUYRES

RESOLUTION D'EQUATIONS DE CONVOLUTION PAR FONCTIONS-SPLINE POLYNOMIALES DANS LE CADRE DE DONNEES BRUITEES

Soutenue le 28 juin 1982, devant la Commission d'Examen

MM.	TEMAM	Président
	BOIZIAU	
	GAROT	Examinateurs
	JOURON	

ORSAY
N° d'ordre : 3264

THESE

PRESENTEE

A L'UNIVERSITE DE PARIS-SUD
CENTRE D'ORSAY

POUR OBTENIR

LE TITRE DE DOCTEUR 3ème CYCLE
Spécialité : ANALYSE NUMERIQUE

par

Agnès WIERNIK - MALGOUYRES

RESOLUTION D'EQUATIONS DE CONVOLUTION PAR FONCTIONS-SPLINE POLYNOMIALES DANS LE CADRE DE DONNEES BRUITEES

Soutenue le 28 juin 1982, devant la Commission d'Examen

MM.	TEMAM	Président
	BOIZIAU	
	GAROT	Examinateurs
	JOURON	

UNIV.
LEIDEN
BIBL.

REMERCIEMENTS

Monsieur le Professeur TEMAM, après m'avoir autorisée à suivre le DEA d'analyse numérique et optimisation, m'a permis de bénéficier d'une allocation de recherche DGRST et je lui en suis très reconnaissante ; je lui exprime, par ailleurs, toute ma gratitude pour avoir accepté de patronner cette thèse.

Je remercie Monsieur JOURON qui a bien voulu être membre du jury.

Monsieur MANUS a accepté que cette thèse soit effectuée dans le Service de Physique des Atomes et des Surfaces du Centre d'Etudes Nucléaires de Saclay et je l'en remercie.

Je tiens à remercier Monsieur BOIZIAU qui a suivi avec intérêt mon travail tout au long des deux années que j'ai passée dans son équipe et qui a su me conseiller et m'aider de manière fructueuse. Je le remercie également d'avoir accepté d'être membre du jury.

Madame GAROT a bien voulu m'initier à la méthode d'autodéconvolution par fonction-spline et s'intéresser, par la suite, à l'évolution de mon travail et je l'en remercie. Je lui suis, aussi, reconnaissante d'avoir accepté d'être membre du jury.

Je remercie Monsieur LICHNEWSKY pour les discussion que j'ai pu avoir avec lui et les conseils qu'il m'a donnés.

Je tiens à exprimer ma reconnaissance envers Madame REYNAUD et Monsieur ROUSSEL pour la patience dont ils ont fait preuve en m'exposant les problèmes auxquels ils sont confrontés et leur origine physique.

Enfin, je remercie tous les membres du Service de Physique des Atomes et des Surfaces pour le climat amicale qu'ils ont su créer et la gentillesse qu'ils ont témoignée à mon égard.

Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41543515_0028

TABLE DES MATIERES

I - INTRODUCTION 1

I.1 - Le problème tel qu'il se pose aux physiciens et le but du présent travail 1

I.2 - Nature du signal étudié 2

I.3 - Nature du bruit étudié 6

I.4 - L'approche proposée et son contexte bibliographique 6

II - L'OUTIL MATHÉMATIQUE EMPLOYÉ 9

LES FONCTIONS-SPLINE

II.1 - Pourquoi le choix des fonctions-spline 9

II.2 - Définitions et propriétés des fonctions-spline polynômiales 10

II.3 - Etude de l'espace des fonctions-spline naturelles 15

II.4 - Propriété des fonctions-spline naturelles 16

II.5 - Convolution de deux fonctions-spline naturelles 20

II.6 - Autoconvolution de deux fonctions-spline naturelles 25

II.7 - Les fonctions-spline d'interpolation 26

II.8 - Les fonctions-spline d'ajustement 35

III - LE DÉBRUITAGE DES SPECTRES 39

III.1 - Méthode de débruitage dans le cas d'un bruit gaussien stationnaire 39

- L'emploi des spline d'ajustement
- La méthode de validation croisée
- L'algorithme d'UTRERAS

III.2 - Méthode de débruitage dans le cas d'un bruit gaussien non stationnaire 51

IV - LA DÉCONVOLUTION INSTRUMENTALE 55

IV.1 - Modélisation de la fonction instrument 55
IV.2 - Modélisation de l'information cherchée 58
IV.3 - Modélisation du signal 58
IV.4 - Formulation du problème et résolution 60

V - L'AUTODÉCONVOLUTION 65

V.1 - Modélisation de l'information cherchée 65
V.2 - Modélisation du signal 66
V.3 - Formulation du problème et résolution 67

VI - RÉALISATION ET EXPLOITATION 71

VI.1 - Mise en oeuvre 71

- Le débruitage
- La déconvolution instrumentale
- L'autodéconvolution

VI.2 - Les tests sur des simulations mathématiques.

VII - CONCLUSION 87

Annexe I Le sous-programme DMAL 2 125
Annexe II Le sous-programme DMAGT 1 133
Annexe III Listing de débruitage 141
Annexe IV Listing de déconvolution instrumentale 151
Annexe V Listing d'autodéconvolution 169

I - INTRODUCTION

I.1 - LE PROBLÈME TEL QU'IL SE POSE AUX PHYSICIENS ET LE BUT DU PRÉSENT TRAVAIL

Le Service de Physique des Atomes et des Surfaces (DPhg. S.P.A.S.) et plus précisément le groupe qui s'intéresse aux interactions "atomes métastables-surfaces", se trouve fréquemment confronté à des traitements de données bruitées. En effet, les valeurs expérimentales sont entâchées d'une incertitude statistique due au fait qu'elles sont obtenues par une méthode de comptage.

L'objet de ce travail est de chercher à extraire les meilleures informations possibles des spectres recueillis par les physiciens. C'est pourquoi nous sommes amenés à construire une fonction "modèle" qui approxime, le mieux possible, une fonction inconnue.

Pour ce faire, il est nécessaire de déterminer correctement l'espace E où l'on choisira de chercher la fonction approximante. Les critères dont nous aurons à tenir compte sont les suivants :

1 - L'espace E ne doit pas être choisi trop grand car sinon on risque d'obtenir, pour de faibles variations sur les données, une grande instabilité sur les solutions.

2 - Les éléments de E doivent être des fonctions d'allure générale "lisse". Ceci est toujours souhaitable quand on a à

traiter des données expérimentales physiques, et d'autant plus indispensable quand ces données proviennent du résultat d'une opération de convolution (ce qui, comme nous le verrons par la suite, est le cas de la majorité des spectres auxquels nous nous sommes intéressés).

3 - Les éléments de E doivent avoir une forme facilement utilisable pour le calcul par ordinateur.

4 - L'espace E doit être suffisamment vaste pour que l'on puisse approximer, de manière satisfaisante, les fonctions auxquelles on a à faire.

Il convient, avant tout, d'analyser les spectres qui nous sont proposés. Nous écrirons schématiquement :

Spectre = Signal + Bruit

I.2 - NATURE DU SIGNAL ÉTUDIÉ

Il faut distinguer trois sortes de signaux différents :

a) Le signal est l'image directe de l'information cherchée :

Signal = information cherchée

b) Le signal est le produit de convolution de l'information cherchée par une fonction-instrument :

Signal = information cherchée * fonction-instrument

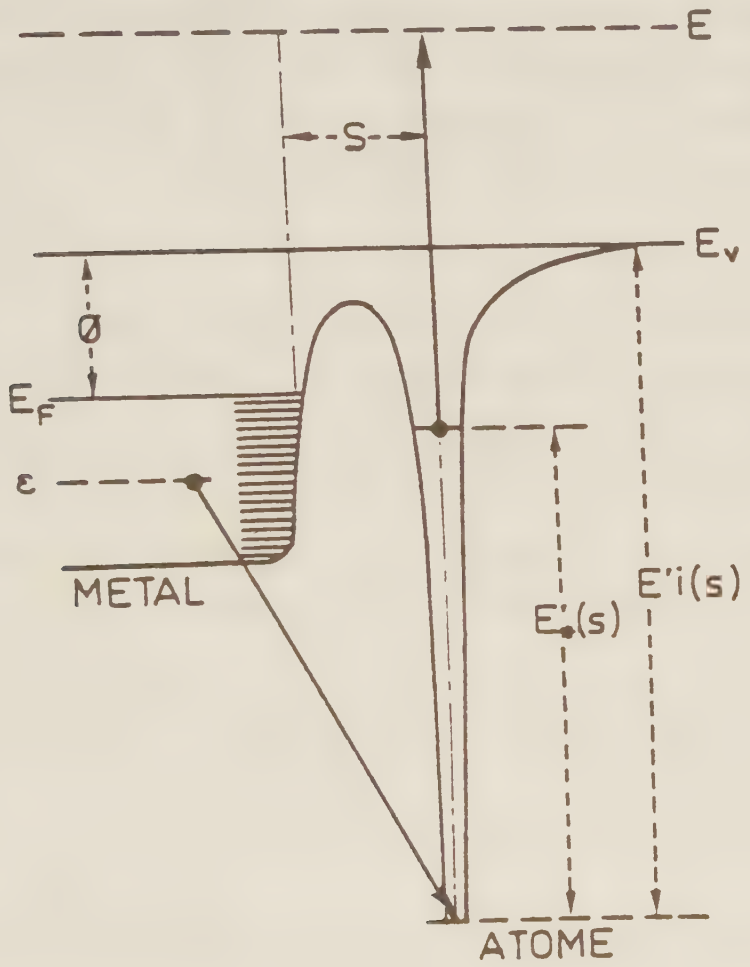
c) Le signal est le produit d'autoconvolution de l'information cherchée par elle-même :

Signal = information cherchée * information cherchée

I.2.a - Signal = information cherchée

C'est le cas, par exemple, où l'on obtient par spectroscopie de désexcitation d'atomes métastables (M.D.S.) avec processus de désexcitation Auger directe (A.D.), l'image directe de la densité d'état de la surface étudiée (cf. C. BOIZIAU / 1 /).

Une telle expérience suit le schéma suivant :



Soient :

. f la représentation mathématique du signal (connu) ;

. u la représentation mathématique de l'information cherchée.

On considère que, dans ce cas-là, la relation qui lie u et f s'écrit :

$$f(x) = u(x)$$

I.2.b - Signal = information cherchée * fonction-instrument

C'est le problème dit de "déconvolution instrumentale" que l'on rencontre, soit lorsque l'expérience est faite avec un instrument dont la "fonction-instrument" n'est pas négligeable, soit lorsque l'expérience est effectuée par "temps de vol" (cf. F. MERCIER / 8 /) et dans ce cas, on parle de "fonction-porte" et non plus de "fonction-instrument".

Soient :

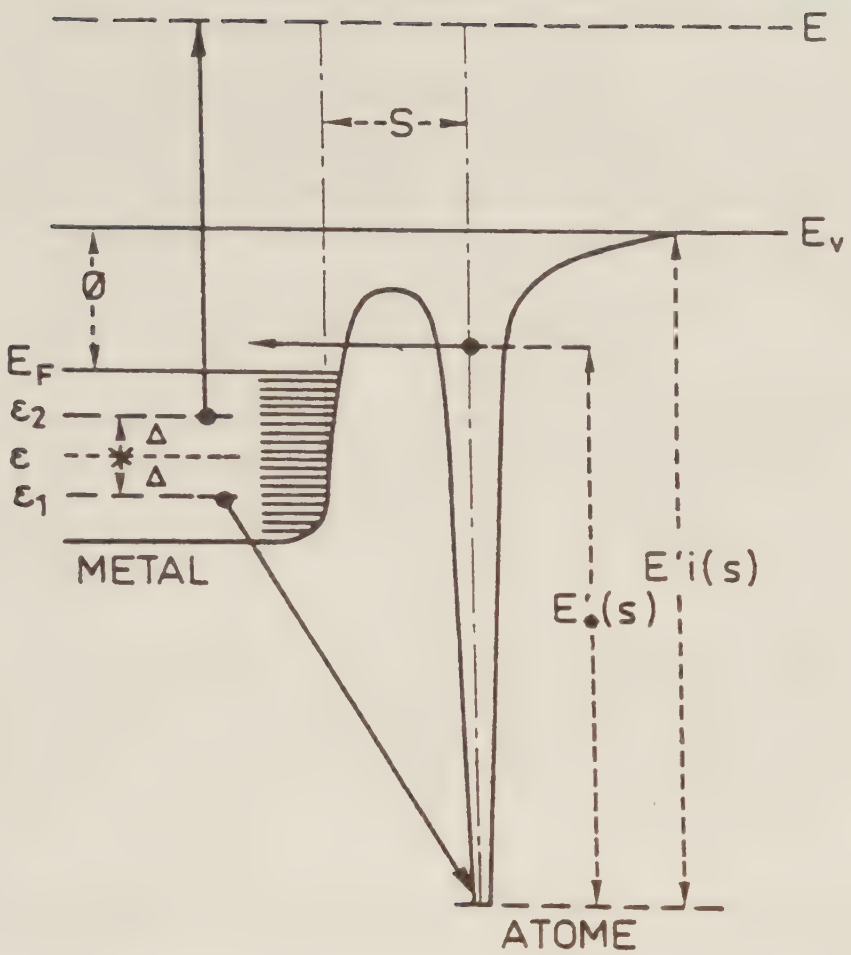
. f la représentation mathématique du signal (connu) ;
. g la représentation mathématique de la fonction-instrument (connue) ;
. u la représentation mathématique de l'information cherchée.

On considère que, dans le cas de la déconvolution instrumentale, la relation qui lie u et f s'écrit :

$$f(x) = \int u\,(x-t)\; g\;(t)\; dt$$

I.2.c - Signal = information cherchée * information cherchée

Ce type de signal se rencontre, entre autres, dans le cas de désexcitation d'atomes métastables (M.D.S.) avec processus d'ionisation résonnante et neutralisation Auger (RI + AN), dont le schéma est le suivant (cf. C. BOIZIAU / 1 /) :



Soient :

. f la représentation mathématique du signal (connu) ;
. u la représentation mathématique de l'information cherchée.

On a alors la relation suivante entre u et f :

$$f(x) = \int u\,(x-t)\;u\,(t)\;dt$$

I.3 - NATURE DU BRUIT ÉTUDIÉ

Les spectres étant obtenus par comptage, les valeurs expérimentales sont les réalisations de variables aléatoires qui suivent des lois de POISSON. Pour les valeurs $f(x_i)$, suffisamment grande, ces lois sont très proches des lois de GAUSS d'espérance $(f(x_i))^{1/2}$ et de variance $f(x_i)$.

C'est-à-dire que si l'on pose :

$$y_i = f(x_i) + \varepsilon_i$$

avec :

y_i = valeur expérimentale au point x_i

$f(x_i)$ = valeur du signal au point x_i

ε_i = "bruit" au point x_i.

Alors chaque ε_i suit une loi de Gauss d'espérance nulle et de variance $\sigma_i^2 = f(x_i)$:

$$\varepsilon_i \sim \mathcal{N}(o, f(x_i))$$

I.4 - L'APPROCHE PROPOSÉE ET SON CONTEXTE BIBLIOGRAPHIQUE

I.4.a - Le débruitage

Nous employons ici une méthode de débruitage par fonctions - spline. De larges études ont été effectuées à ce sujet par C.M. REINSCH / 11, 12 /, P. CRAVEN et G. WAHBA / 4 /, G. KIMELDORF et G. WAHBA / 6 /, G. WAHBA et G. WOLD / 24 / et G. WAHBA / 21, 22, 23 /.

Dans sa thèse, F. UTRERAS / 17, 18, 19, 20 / a développé un algorithme, fondé sur une idée de G. WAHBA / 23 /, qui permet de débruiter de façon optimale un spectre de valeurs expérimentales entâchées d'une incertitude statistique gaussienne de variance constante. C'est ce travail qui a servi de base à la méthode exposée ici.

I.4.b - La déconvolution instrumentale

De nombreuses techniques ont déjà été utilisées pour résoudre le problème de la déconvolution instrumentale, et on trouvera une bonne bibliographie à ce sujet dans la thèse de F. MERCIER / 8 /.

Notre propos, ici, a été de mettre en oeuvre une méthode de déconvolution instrumentale par fonctions - spline polynomiales (par analogie avec la méthode proposée par l'autodéconvolution).

I.4.c - L'autodéconvolution

Sur le plan de l'autodéconvolution ce travail fait suite à celui de C. GAROT / 5 /, à qui revient l'idée originale d'employer des fonctions-spline pour résoudre des équations d'autoconvolution. On trouvera dans sa thèse un résumé des avantages et des inconvénients des autres méthodes d'autodéconvolution, ainsi que la description des phénomènes physiques nécessitant, pour être exploitables, un calcul d'autodéconvolution.

I.4.d - Les fonctions-spline

Les ouvrages sur les fonctions-spline sont nombreux. Historiquement, le premier est celui de I.J. SCHOENBERG / 13 /. Ceux qui nous ont servi de base ici, sont :

- celui de P.J. LAURENT / 7 /, qui présente une étude hibertienne des fonctions-spline et propose des méthodes de calcul que nous utiliserons,
- celui de L.L. SCHUMAKER / 14 /, qui donne un très large exposé sur la théorie générale des fonctions-spline,
- celui de M.J.D. POWELL / 10 /, qui étudie l'utilisation des fonctions-spline dans le cadre de l'approximation de données, et enfin,
- celui de C. DE BOOR / 2 /, qui fait le point sur l'état actuel des connaissances concernant les fonctions-spline.

II - L'OUTIL MATHEMATIQUE EMPLOYE
LES FONCTIONS-SPLINE

II.1 - POURQUOI LE CHOIX DES FONCTIONS-SPLINE ?

Historiquement, le terme de "fonction-spline" a été utilisé pour la première fois par SCHOENBERG / 13 / pour désigner des fonctions formées de morceaux de polynômes dont les dérivées se raccordent jusqu'à un certain ordre en chaque point de jonction.

On obtient notamment des fonctions-spline en cherchant à minimiser, pour $q \in \mathbb{N}$ fixé, la fonctionnelle :

$$\phi(f) = \int_a^b \left[D^q f(t)\right]^2 dt$$

$$a \in \mathbb{R} \qquad b \in \mathbb{R}$$
$$a < b$$

sur l'ensemble des fonctions qui vérifient des conditions du type :

$$D^j (f) (t_i) = y_{ij}$$

(où les t_i représentent les points de jonction des morceaux de polynômes).

On voit donc qu'en jouant sur ces conditions et sur la valeur de q, on arrive à construire des fonctions dotées d'un pouvoir lissant (cette qualité sera particulièrement intéressante pour le débruitage).

D'autre part, le fait qu'elles soient construites à partir de polynômes, nous assurent un usage facile de ces fonctions sur ordinateur et le calcul aisé de leurs dérivées (qui sont d'ailleurs elles aussi des fonctions-spline).

Enfin, pour ce qui nous concerne, un avantage des fonctions-spline est qu'elles forment un espace dans lequel la convolution est une loi interne et que, comme l'a démontré C. GAROT / 5 /, l'autoconvolution est injective de l'espace des fonctions-spline à valeurs positives dans l'espace des fonctions-spline autoconvolées.

II.2 - DÉFINITIONS ET PROPRIÉTÉS DES FONCTIONS-SPLINE POLYNOMIALES

Définition 1 - Espace des fonctions-spline polynomiales

Soit $[a,b] \subset \mathbb{R}$ et $q \in \mathbb{N}$ fixé

soit $\Delta(a,b) = \{ t_i \}_{i = 1,2 \ldots K}$

Une subdivision de $[a,b]$

avec $a = t_o < t_1 < \ldots < t_K < t_{K+1} = b$

et soit $\mathcal{M} = (m_1\ m_2 \ldots m_K)$ K

avec $1 < m_i < q$ $\forall\ i = 1,2 \ldots K$

On pose :

$\mathcal{S}(q,\Delta(a,b),\mathcal{M}) = \{$ S telles qu'il existe des polynômes $p_o\ p_1 \ldots p_K\ p_{q-1}$ avec :

$$1)\ S(x) = p_i(x) \quad \text{pour } x \in [t_i, t_{i+1}[\quad (i = 0,1,2,\ldots K)$$

$$2)\ D^j p_{i-1}(t_i) = D^j p_i(t_i) \quad i = 1,2,\ldots K \quad j = 0,1,\ldots, q-1-m_i$$

$\mathcal{S}(q, \Delta(a,b), \mathcal{M})$ est l'espace des fonctions-spline polynômiales d'ordre q pour les noeuds $(t_1, t_2, \ldots, t_K)$ de multiplicité $(m_1, m_2, \ldots, m_K)$.

Définition 2 - Vecteur de multiplicité

Le vecteur $\mathcal{M} = (m_1\ m_2 \ldots m_K) \in \mathbb{R}^K$ est le vecteur de multiplicité des noeuds $(t_1, t_2 \ldots t_K)$.

Remarque 1

Du vecteur $\mathcal{M}$ (vecteur de multiplicité) dépend le caractère plus ou moins lisse de la fonction-spline polynomiale en chaque noeud.

Théorème 1

Soit $KM = \sum_{i=1}^{K} m_i$

$\mathcal{S}(q, \Delta(a,b), \mathcal{M})$ est un espace vectoriel de dimension q + KM.

Définition 3 - Les fonctions-spline à noeuds simples

Si $m_i = 1$ pour $i = 1, 2 \ldots K$

Nous noterons $\mathcal{S}(q, \Delta(a,b), \mathcal{M}) = S(q, \Delta(a,b))$

S (q, Δ (a,b)) est l'espace des fonctions-spline polynômiales d'ordre q avec noeuds simples.

Remarque 2

$S(q, \Delta(a,b))) \subset \mathcal{C}^{q-2}(a,b)$

Remarque 3

$S(q, \Delta(a,b))$ est un espace vectoriel de dimension $q + K$.

Cas particulier important des fonctions-spline polynômiales d'ordre pair

Posons : $q = 2p$

Nous venons de voir que $S(2p, \Delta(a,b))$ est un espace vectoriel de dimension $2p + K$.

Anticipons sur la suite en supposant que nous imposons à une fonction-spline polynômiale s les K relations d'interpolation suivantes :

$$Y = (y_1, y_2, \ldots, y_K) \in \mathbb{R}^K \text{ fixé}$$

$$(r_i) = y_i \qquad i = 1, 2 \ldots K$$

La fonction-spline s possède alors 2p degrés de liberté que l'on détermine, par exemple, en choisissant de prendre :

$$s\Big|_{[a,t_1[} \in \mathbb{P}_{p-1}$$

et

$$s\Big|_{[t_K,b[} \in \mathbb{P}_{p-1}$$

Ce qui nous conduit à une nouvelle définition:

Définition 4 - Espace des fonctions spline naturelles

Soit :

$$SN\ (q,\ \Delta\ (a,b))) = \Big\{\ s \in S\ (2q,\ \Delta\ (a,b))$$

telles que

$$S \times \chi_{[a,t_1[} \in \mathbb{P}_{q-1}$$

et

$$S \times \chi_{[t_{K,b}[} \in \mathbb{P}_{q-1}\ \Big\}$$

avec $$\chi_{[\alpha,\beta[}(x) = \begin{cases} 1 & \text{si } \alpha \leqslant x < \beta \\ 0 & \text{sinon} \end{cases}$$

SN (q, Δ (a,b)) est l'espace des fonctions-spline naturelles d'ordre q.

Théorème 2

SN (q, Δ (a,b)) est un espace vectoriel de dimension K.

Remarque 4

$$SN\ (q,\ \Delta\ (a,b)) \subset \mathcal{C}^{2q-2}\ (a,b)$$

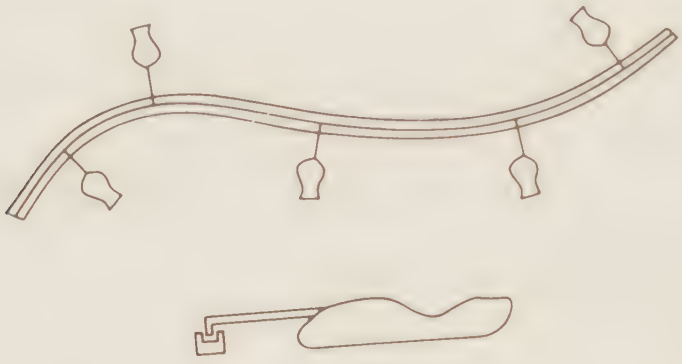
Etude "historique" de l'espace des fonctions-spline naturelles

Considérons l'espace SN (2, Δ (a,b)) (appelé espace des fonctions-spline naturelles cubiques ou de façon plus concise espace des fonctions-spline cubiques)

$$SN\ (2,\ \Delta\ (a,b)) \subset \mathcal{C}^{2}(a,b)$$

Donc SN (2, Δ (a,b)) est l'ensemble des fonctions bicontinûment différentiables formées de morceaux de polynômes de degré 1 sur les intervalles $[a, t_1[$ et $]t_K, b]$ et de morceaux de polynômes de degré 3 sur les intervalles $[t_i, t_{i+1}[$ pour i = 1,2 ... K-1.

Or, le terme "spline" désigne, en anglais, un instrument utilisé par les architectes pour dessiner des courbes pssant par des points fixés :



Et, vers le milieu du 18ème siècle, EULER et les frères BERNOUILLI ont découvert que la position d'équilibre d'une telle latte est approximativement donnée par des morceaux de polynômes de degré 3 reliés aux points de jonction par leurs dérivées jusqu'à l'ordre 2.

Et si on lâche la latte aux deux points fixés extrêmes ; on imagine bien que, naturellement, elle prendra l'allure d'un morceau de droite sur les deux extrémités. De là vient le nom de "fonction-spline naturelle " donné aux éléments de l'espace SN (q, Δ (a,b)).

Par la suite, nous nous intéresserons plus précisément à cet espace SN (q, Δ (a,b)) dont les éléments peuvent se mettre sous une forme particulièrement agréable à manipuler, comme nous allons le voir dans le paragraphe suivant.

II.3 - ÉTUDE DE L'ESPACE DES FONCTIONS-SPLINE NATURELLES

Définition pratique de SN (q, Δ (a,b))

SN (q, Δ (a,b)) est l'espace des fonctions réelles s définies sur $[a,b]$ telles que :

1. s est un polynôme de degré 2q-1 dans chacun des intervalles $[t_i, t_{i+1}[$ pour $i = 1, 2, \ldots K-1$.
2. s est un polynôme de degré q-1 dans $[a,t_1[$ et $]t_K,b]$.
3. D^{2q-2} s est continue.

où les $(t_i)_{i = 1,2 \ldots K}$ avec $K > q$ représentent K abscisses distinctes de l'intervalle $[a,b]$ telles que :

$$a < t_1 < t_2 \ \ldots \ < t_K < b$$

$$\text{et} \quad \Delta (a,b) = \{t_i\}_{i = 1, 2, \ldots K}$$

Remarque

SN (q, Δ (a,b)) forme un sous-espace vectoriel dans H^q (a,b)

où $H^q (a,b) = \{f : [a,b] \to \mathbb{R}$ telles que

$$f \in \mathcal{C}^{q-1}(a,b)$$

$$\text{et} \quad \int_a^b \left(D^q (f) (t)\right)^2 dt < \infty$$

H^q (a,b) est un espace de Hilbert pour le produit scalaire :

$$< f,g >_q = \sum_{i=o}^{q} \int_a^b D^i f(t) \, D^i g(t) \, dt$$

et la norme associée $||f||_q = (< f,f >_q)^{1/2}$.

II.4 - PROPRIÉTÉS DES FONCTIONS-SPLINE NATURELLES

Notation

$$(e)_+ = \max (e, o)$$

Proposition fondamentale

$$S \in SN\ (q, \Delta(a,b)) \iff \boxed{\begin{array}{l} S(t) = \sum_{j=o}^{q-1} \alpha_j\, t^j + \sum_{i=1}^{K} \mu_i\, \frac{(t-t_i)_+^{2q-1}}{(2q-1)\ !} \\ \qquad\qquad \text{sur} \quad a,b \\ \text{avec} : \sum_{i=1}^{K} \mu_i\, (t_i)^n = 0 \\ \qquad\qquad \text{pour } m = 0, \ldots, q-1 \end{array}}$$

Démonstration

a) Soit :

$$\delta_i\ (t) = \frac{(t - t_i)_+^{2q-1}}{(2q-1)\ !}$$

δ_i a une discontinuité de sa dérivée $(2q-1)^{\text{ème}}$ égale à 1 en t_i, et toutes ses autres dérivées sont continues et égales à 0 en t_i.

Soit p_i le polynôme de degré 2q-1 tel que :

$$p_i\ (t) = s(t) \cdot \chi_{[t_i,\ t_{i+1}[}(t)$$

alors on obtient par développement de Taylor :

$$p_i\ (t) = p_{i-1}(t) + \mu_i\ \delta_i\ (t)$$

$$\text{pour} \quad t \in [t_i,\ t_{i+1}[$$

où d_i est la valeur de discontinuité de $D^{2q-1}(s)$ en t_i

$$d_i = D^{3q-1}(s)(t_i^+) - D^{2q-1}(s)(t_i^-)$$

donc V s SN (q, Δ (a,b))

$$s(t) = p_o(t) + \sum_{i=1}^{K} \mu_i \frac{(t - t_i)_+^{2q-1}}{(2q-1)\ !}$$

où p_o est un polynôme de degré q-1.

b) Réciproquement, toute fonction de la forme :

$$p_o(t) + \sum_{i=1}^{K} \mu_i \frac{(t - t_i)_+^{2q-1}}{(2q-1)\ !} = s(t)$$

n'appartient pas obligatoirement à SN (q, Δ (a,b)).

En effet, si les conditions 1 et 3 de la définition de la page 14 sont vérifiées de façon évidente, il n'en est pas de même de la condition 2 .

Pour que ce soit le cas, il faut que s soit un polynôme de degré q-1 sur $]t_K, b]$, c'est-à-dire qu'il faut que

$$\sum_{i=1}^{K} \mu_i \frac{(t - t_i)^{2q-1}}{(2q-1)\ !}$$

soit un polynôme de degré q-1.

Par conséquent, il faut que :

$$\sum_{i=1}^{K} \mu_i \frac{(t - t_i)^{q-1}}{(q-1)\ !} = 0$$

En exprimant que les dérivées successives de ce polynôme sont nulles en zéro on obtient :

$$\sum_{i=1}^{K} \mu_i \, (t_i)^n = 0$$

pour m = 0, 1,... q-1

Remarque

Les contraintes sur les paramètres :

$$\sum_{i=1}^{K} \mu_i \, (t_i)^m = 0$$

pour m = 0, 1,,... q-1

sont indispensables pour assurer le caractère lissant d'une fonction-spline naturelle à qui l'on impose des conditions d'interpolation.

Contre-Exemple

Soit le vecteur $Y = (y_i)_{i= 1,20}$ défini comme suit :

$$y_i = 0 \qquad \text{si } i \neq 3$$

$$y_3 = 10$$

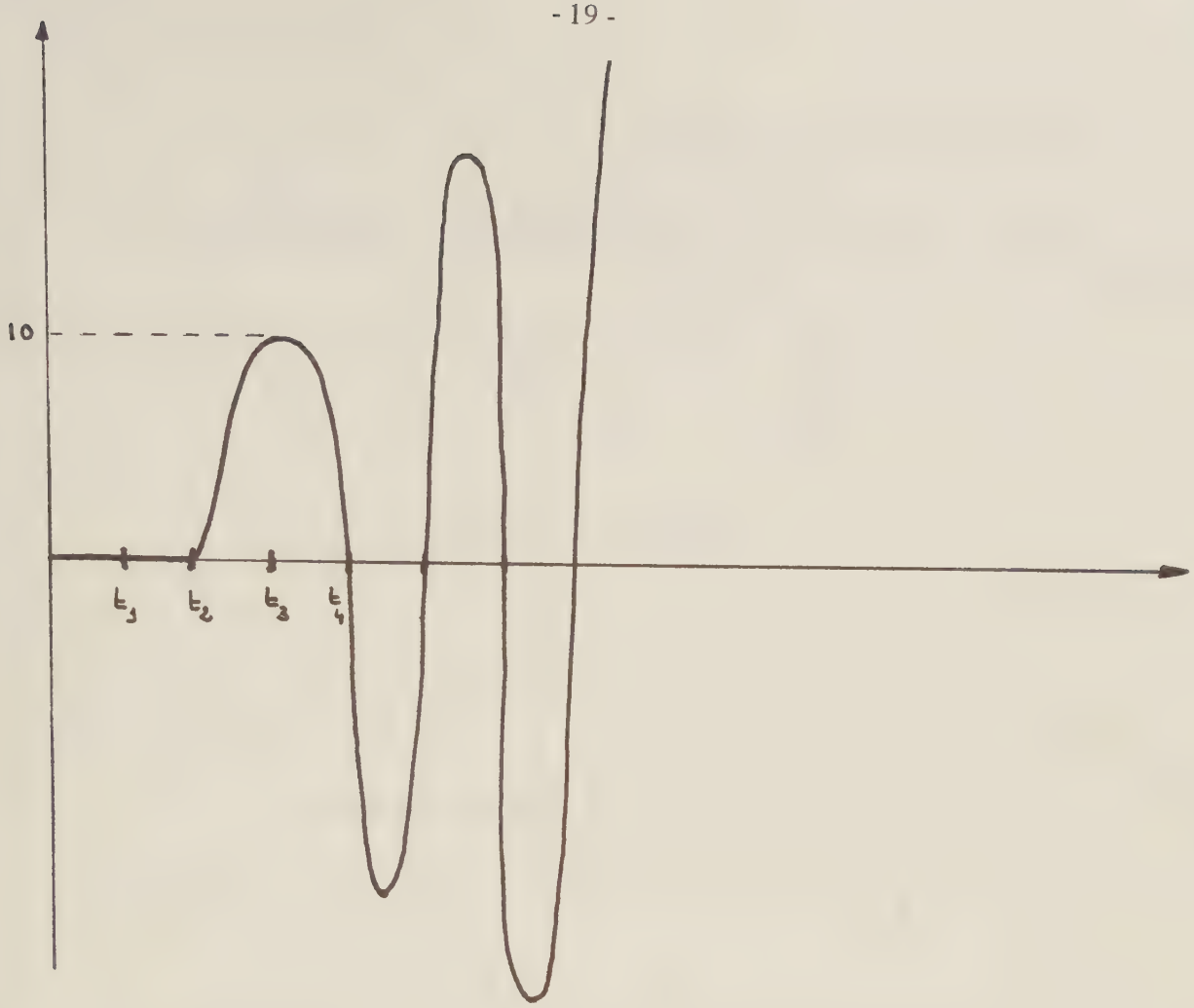
On calcule les coefficients $(b_i)_{i= 1, 20}$ du polynôme :

$$P(t) = b_1 + b_2 t + \sum_{1=1}^{18} b_{2+1} \frac{(t - t_1)_+^3}{3\,!}$$

qui vérifie :

$$P(t_i) = y_i \qquad i = 1, 20$$

et on obtient un polynôme qui a l'allure suivante :



Alors que si l'on calcule les coefficients $(b_i)_{\ i = 1,\ 20}$ du polynôme :

$$P(t) = b_1 + b_2 t + \sum_{1=1}^{18} b_{2+1} \frac{(t - t_1)_+^3}{3\ !}$$

qui vérifie :

$$P(t_i) = y_i \qquad i = 2,\ 19$$

avec les conditions :

$$\sum_{1=1}^{18} b_{2+1}\ (t_1)^m \qquad m = 0,\ 1$$

on obtient un polynôme qui a l'allure suivante :

Cas des fonctions-spline cubiques

Pour le cas où q=2, nous adopterons les notations suivantes :

$$s(t) = b_1 + b_2 t + \sum_{1=1}^{K} b_{2+1} \frac{(t - t_1)_+^3}{3\ !}$$

avec $\sum_{1=1}^{K} b_{2+1} = 0$

$\sum_{1=1}^{K} t_1\, b_{2+1} = 0$

II.5 - CONVOLUTION DE DEUX FONCTIONS-SPLINE NATURELLES

Soient $[a,b]$ et $[c,d]$ deux intervalles de $\mathbb{R}$ avec :

. $\Delta_1 (a,b) = \{t_i\}_{i=1,NT}$ une subdivision de (a,b)

avec $a = t_o < t_1 < \ldots < t_{NT} < t_{NT+1} = b$

. $\Delta_2 (c,d) = \{r_i\}_{i=1,NR}$ une subdivision de (c,d)

avec $c = r_o < r_1 < \ldots < r_{NR} < r_{NR+1} = d$

Considérons :

s_1 SN $(q, \Delta_1 (a,b))$ $q \in \mathbb{N}$

et

s_2 SN $(p, \Delta_2 (c,d))$ $p \in \mathbb{N}$

et désignons par S leur produit de convolution :

$$\boxed{S(x) = \int_{\mathbb{R}} s_2\ (x-t)\ s_1\ (t)\ dt}$$

d'après la proposition fondamentale (cf. § II.4)

$$s_1\ (t) = \sigma_1\ (t)\ \chi_{(a,b)}\ (t)$$

$$s_2\ (x-t) = \sigma_2\ (x-t)\ \chi_{(c,d)}\ (x-t)$$

où :

$$\sigma_1(t) = \sum_{j=o}^{q-1} \alpha_j\ t^j + \sum_{i=1}^{NT} \mu_i\ \frac{(t - t_i)_+^{2q-1}}{(2q-1)\ !}$$

$$\text{avec} \quad \sum_{i=1}^{NT} \mu_i\ (t_i)^K = 0 \qquad K = 0, \ldots, q-1$$

et

$$\sigma_2(t) = \sum_{m=o}^{p-1} \beta_m\ t^m + \sum_{1=1}^{NR} \nu_1\ \frac{(t - r_1)_+^{2p-1}}{(2p-1)\ !}$$

$$\text{avec} \quad \sum_{i=1}^{NR} \nu_1\ (t_i)^K = 0 \qquad K = 0, \ldots; q-1$$

donc

$$S(x) = \int_{\mathbb{R}} \sigma_1\ (t)\ \sigma_2\ (x-t)\ \chi_{a,b}\ (t)\ \chi_{c,d}\ (x,t)\ dt$$

c'est-à-dire que :

$$\boxed{S(x) = \int_{\sup\ (a,\ x-d)}^{\inf\ (b,\ x-c)} \sigma_1\ (t)\ \sigma_2\ (x-t)\ dt}$$

Les signaux que nous avons à étudier représentent un nombre d'électrons par pas unitaire de l'échelle des énergies et sont donc à valeurs positives ; pour cette raison, nous ne nous intéresserons par la suite qu'au cas :

$$a = 0 \qquad \text{et} \qquad c = 0$$

ce qui donne :

$$S(x) = \int_{\sup(o,x-d)}^{\inf(b,x)} \sigma_1 (t)\ \sigma_2 (x-t)\ dt$$

Supposons, par exemple que $\boxed{b < d}$ nous pouvons alors exprimer S(x) de la façon suivante :

- <u>1er cas</u> $x < b$

$$S(x) = \int_o^x \sigma_1 (t)\ \sigma_2 (x-t)\ dt$$

- <u>2ème cas</u> $b < x < d$

$$S(x) = \int_o^b \sigma_1 (t)\ \sigma_2 (x-t)\ dt$$

- <u>3ème cas</u> $d < x < b + d$

$$S(x) = \int_{x-d}^b \sigma_1 (t)\ \sigma_2 (x-t)\ dt$$

De ces trois formules, celle qui conduit à des résultats facilement utilisables est la première, on a alors :

$$S(x) = \sum_{j=o}^{q-1} \sum_{m=o}^{p-1} \alpha_j \beta_m \int_o^x t^j (x-t)^m \, dt$$

$$+ \sum_{j=o}^{q-1} \sum_{l=1}^{NR} \alpha_j \nu_l \int_o^x t^j \frac{(x-t-r_l)_+^{2p-1}}{(2p-1)!} \, dt$$

$$+ \sum_{i=1}^{NT} \sum_{m=o}^{p-1} \mu_i \beta_m \int_o^x \frac{(t-t_i)_+^{2q-1}}{(2q-1)!} (x-t)^n \, dt$$

$$+ \sum_{i=1}^{NT} \sum_{l=1}^{NR} \mu_i \nu_l \int_o^x \frac{(t-t_i)_+^{2q-1}}{(2q-1)!} \frac{(x-t-r_l)_+^{2p-1}}{(2p-1)!} \, dt$$

Ce qui donne, en intégrant par parties :

$$S(x) = \sum_{j=o}^{q-1} \sum_{m=o}^{p-1} \alpha_j \beta_m \frac{m!\, j!}{(j+m+1)!} x^{j+m+1}$$

$$+ \sum_{j=o}^{q-1} \sum_{l=1}^{NR} \alpha_j \nu_l \frac{j!}{(j+2p)!} (x-r_l)_+^{2p+j}$$

$$+ \sum_{i=1}^{NT} \sum_{m=o}^{p-1} \mu_i \beta_m \frac{m!}{(m+2q)!} (x-t_i)_+^{2q+m}$$

$$+ \sum_{i=1}^{NT} \sum_{l=1}^{NR} \mu_i \nu_l \frac{1}{(2q+2p-1)!} (x-t_i-r_l)_+^{2q+2p-1}$$

Définissons alors une nouvelle subdivision de l'intervalle (o, b) :

$$\Delta_3\ (o,b) = \left\{ \{t_i\}_{i=1,NT} \cup \{r_1\}_{1=1,NR} \cup \{t_i+r_1\}_{\substack{i=1,NT \\ 1=1,NR}} \right\} \cap [o,b]$$

$$= \{\tau_j\}_{j=1,NS}$$

Nous venons de démontrer que :

1 - S est un polynôme de degré p+q+1 sur l'intervalle (o, τ_1)

2 - S est un polynôme de degré 2p+2q-1 sur chacun des intervalles $[\tau_i, \tau_{i+1}[$

3 - $D^\gamma S$ est continue

avec $\gamma = \inf (2p + q-2, 2q + p-2)$

En conclusion, nous pouvons énoncer le théorème suivant :

Théorème 3

Soient : $0 < A < B < +\infty$, $q \in \mathbb{N}$, $p \in \mathbb{N}$

$S_1 \in SN\ (q, \Delta_1\ (O,A))$ où $\Delta_1\ (O,A) = \{t_i\}_{i=1,NT}$

$S_2 \in SN\ (p, \Delta_2\ (O,B))$ où $\Delta_2\ (O,B) = \{r_i\}_{i=1,NR}$

et soit

$$S(x) = \int_o^x S_1(t)\ S_2(x-t)\ dt \qquad \text{pour } x \in [O,A]$$

Alors

$$S \in (2p + 2q, \Delta_3\ (O,A), \mathcal{U}_*)$$

où

$$\Delta_3\ (0,A) = \left\{ \{t_i\}_{i=1,NT} \cup \{r_1\}_{1=1,NR} \cup \{t_i+r_1\}_{\substack{i=1,NT \\ 1=1,NR}} \right\} \cap [0,A]$$

$$= \{\tau_j\}_{j=1,NS}$$

et où $\mathcal{M}_*$ est le vecteur de multiplicité de convolution défini comme suit :

$$\mathcal{M}_* = (m_i^*)_{i=1,NS}$$

avec :

$m_i^* = 2p$ si $\tau_i \in \Delta_1\ (0,A)$

$m_i^* = 2q$ si $\tau_i \in \Delta_2\ (0,B)$

$m_i^* = 1$ si $\tau_i \in \Delta_3\ (0,A) \setminus \Delta_1\ (0,A) \setminus \Delta_2\ (0,B)$

II.6 - AUTOCONVOLUTION DE DEUX FONCTIONS-SPLINE NATURELLES

C. GAROT / 5 / a démontré le théorème d'unicité suivant :

Théorème 4

Soient $0 < A < +\infty$ et $q \in \mathbb{N}$

$$S_1 \in SN\ (q, \Delta\ (0,A))$$

et

$$S_2 \in SN\ (q, \Delta\ (0,A))$$

où $\Delta\ (0,A)$ est une subdivision de $[0,A]$.

et soient S_1 et S_2 définies sur $[0,A]$ telles que :

$$S_1(x) = \int_0^x s_1(t)\, s_1(x-t)\, dt$$

$$S_2(x) = \int_0^x s_2(t)\, s_2(x-t)\, dt$$

Alors :

$$\forall\, x \in [0,A] \quad S_1(x) = S_2(x) \iff s_1 \equiv s_2 \text{ ou } s_1 \equiv -s_2$$

Remarque

Ce résultat nous assure donc que si l'on connaît une fonction-spline polynômiale naturelle en tant que solution d'une équation d'autoconvolution, on est sûr de son unicité au signe près.

II.7 - LES FONCTIONS-SPLINE D'INTERPOLATION

Lorsque, comme c'est notre cas, on souhaite utiliser les fonctions-spline pour approximer des données ; il convient d'étudier deux sortes de fonctions-spline polynômiales naturelles : Les fonctions-spline d'interpolation et les fonctions-spline d'ajustement.

Les fonctions-spline d'interpolation (dont une étude détaillée est faite dans l'ouvrage de P.J. LAURENT / 7 /) sont construites pour passer exactement par des points fixés. Elles nous seront donc utiles quand nous aurons à faire à des données expérimentales que l'on pourra considérer comme exactes.

Théorème 5

Soit $Y = (y_1, y_2 \dots y_K) \in \mathbb{R}^K$ un vecteur de données d'abscisses $(t_1, t_2 \dots t_K)$

$$\text{avec } a = t_o < t_1 < t_2 \dots < t_K < t_{K+1} = b$$

et soit $q \in \mathbb{N}$ fixé.

Alors il existe une, et une seule, fonction-spline naturelle d'ordre q basée sur les noeuds $(t_1, t_2 \dots t_K)$ qui interpole les données $(y_1\ y_2 \dots y_K)$.

$$\forall\ (y_1 \dots y_K) \in \mathbb{R}^K \quad \exists!\ s_i \in SN\ (q,\ \Delta\ (a,b))$$

$$(\text{où } \Delta\ (a,b) = \left\{ t_i \right\}_{i=1,K})$$

$$S_i\ (t_j) = y_j \qquad j = 1,2 \dots K$$

lemme 1

$$\forall\ f \in H^q\ (a,b)\ ,\ \forall\ s \in SN\ (q,\ \Delta\ (a,b))$$

$$\int_a^b D^q(s)(t)\ D^q(f)(t)\ dt = (-1)^q \sum_{i=1}^{K} \left[D^{2q-1}(s)(t_i^+) - D^{2q-1}(s)(t_i^-) \right] f(t_i)$$

démonstration du lemme 1

$$\int_a^b D^q(s)(t)\ D^q(f)(t)\ dt = \left[D^q(s)(t)\ D^{q-1}(f)(t) \right]_a^b - \int_a^b D^{q+1}(s)(t)\ D^{q-1}(f)(t)\ dt$$

le terme entre crochets est nul car s est un polynôme de degré q-1 sur les deux intervalles extrêmes.

On intègre ainsi (q-1) fois par parties et on obtient :

$$\int_a^b D^q(s)(t)\ D^q(f)(t)\ dt = (-1)^{q-1} \int_a^b D^{2q-1}(s)(t)\ f'(t)\ dt$$

soit

$$\mu_i = D^{2q-1}(s)(t_i^+) - D^{2q-1}(s)(t_i^-)$$

Posons :

$$D^{2q-1}(s)(t) = V_i \qquad \text{pour} \quad t \in [t_i, t_{i+1}[$$

on a d'autre part

$$D^{2q-1}(s)(t) = 0 \qquad \text{pour } t < t_1$$

$$\text{et pour } t > t_K$$

donc en posant : $Z_i = f(t_i)$ $\qquad i = 1, 2 \ldots K$

on obtient :

$$\int_a^b D^q(s)(t)\ D^q(f)(t)\ dt = (-1)^{q-1} \sum_{i=1}^{K-1} V_i \left(f(t_{i+1}) - f(t_i) \right)$$

soit encore :

$$\int_a^b D^q(s)(t)\ D^q(f)(t)\ dt = (-1)^{q-1} \sum_{i=1}^{K-1} V_i\ (Z_{i+1} - Z_i)$$

En posant $V_o = V_K = 0$

on a :

$$\mu_i = V_i - V_{i-1} \qquad \text{pour} \quad i = 1, 2 \ldots K$$

Ce qui conduit à :

$$(-1)^{q-1} \sum_{i=1}^{K-1} V_i \, (Z_{i+1} - Z_i) = (-1)^{q-1} \left(\sum_{i=o}^{K-1} V_i \, Z_{i+1} - \sum_{i=1}^{K-1} V_i \, Z_i \right)$$

$$= (1)^{q-1} \left(\sum_{i=1}^{K} V_{i-1} \, Z_i - \sum_{i=1}^{K} V_i \, Z_i \right)$$

$$= (-1)^q \sum_{i=1}^{K} (V_i - V_{i-1}) \, Z_i$$

donc, finalement

$$\int_a^b D^{(q)}(s)(t) \, D^{(q)}(f)(t) \, dt = (-1)^q \sum_{i=1}^{K} \mu_i \, Z_i$$

et par conséquent

$$\int_a^b D^q(s)(t) \, D^q(f)(t) \, dt = (-1)^q \sum_{i=1}^{K} D^{2q-1}(s)(t_i^+) - D^{2q-1}(s)(t_i^-) \, f(t_i)$$

Démonstration du théorème 5

Posons :

$$S_i(t) = \sum_{j=o}^{q-1} \alpha_j \, t^j + \sum_{1=1}^{K} \mu_1 \, \frac{(t - t_1)_+^{2q-1}}{(2q-1)\,!}$$

Cette expression dépend de K+q paramètres.

Etant donné que $s_i \in SN\ (q,\ \Delta\ (o,b))$, nous pouvons écrire q relations linéaires

$$\sum_{l=1}^{K} \mu_l\ (t_l)^m = 0 \qquad m = 0,\ 1,\ \ldots\ q-1$$

D'autre part les conditions d'interpolation que l'on impose fournissent K autres relations linéaires entre les paramètres

$$S_i\ (t_j) = y_j \qquad j = 1,\ 2,\ \ldots\ K$$

On a donc un système linéaire de K + q équations à K + q inconnues.

Ce système admettra une solution unique si le système homogène correspondant admet comme seule solution la solution identiquement nulle.

Soit σ_o la solution du problème homogène associé :

$$\sigma_o \quad SN\ (q,\ \Delta\ (a,b))$$

$$\sigma_o\ (t_i) = 0 \qquad \forall\ i = 1,\ 2,\ \ldots\ K$$

Ecrivons le lemme 1 en posant :

$$s = f = \sigma_o$$

On obtient :

$$\int_a^b D^q(\sigma_o)(t)\ D^q(\sigma_o)(t)\ dt = (-1)^q \sum_{i=1}^{K} (D^{2q-1}\ (\sigma_o)(t_i^+) - D^{2q-1}\ (\sigma_o)(t_i))\ \sigma_o\ (t_i)$$

et par suite :

$$\int_a^b \left(D^q (\sigma_o)(t)\right)^2 dt = 0$$

donc σ_o est un polynôme de degré q-1 et comme :

$$\sigma_o (t_j) = 0 \qquad \text{pour } j = 1, 2 \ldots K \text{ avec } K > q$$

on a finalement

$$\sigma_o \equiv 0$$

Définition 5 - Fonctions-spline d'interpolation

L'unique solution s_i du problème :

$(t_1,y_1), (t_2,y_2) \ldots (t_K,y_K)) \in (\mathbb{R}^2)^K$ fixé ; q fixé

$$a < t_i < t_{i+1} < b \qquad i = 1, 2, \ldots K-1$$

$$S (t_i) = y_i \qquad i = 1, 2, \ldots K$$

$$S \in SN (q, \Delta (a,b)) \qquad \text{où } \Delta (a,b) = \{t_i\}_{i=1,K}$$

est appelée fonction-spline d'interpolation (d'ordre q) des données $(y_1\ y_2 \ldots y_K)$.

Propriétés extrêmales - Pouvoir lissant des fonctions-spline d'interpolation

Soit $I_y = \left\{ f \in H^q (a,b) \ / \ f (t_i) = y_i \ , \quad i = 1, 2 \ldots K \right\}$

avec $y = (y_1\ y_2 \ldots y_K) \in \mathbb{R}^K$ fixé.

Théorème 6

Soit $((t_1,y_1)\ (t_2,y_2)\ \ldots\ (t_K,y_K)) \in (\mathbb{R}^2)^K$ fixé

$(a = t_o < t_1 < t_2 \ldots < t_K < t_{K+1} = b)$ et $q \in \mathbb{N}$;

et soit S_i l'unique élément de $SN\ (q, \Delta\ (a,b))$

(où $\Delta\ (a,b) = \{t_i\}_{i=1,K}$) tel que :

$$S_i\ (t_j) = y_j \qquad\qquad j = 1,\ 2 \ldots K$$

Alors :

1°- $\forall\ f \in I_y$

$$\int_a^b \left(D^q(s_i)(t) - D^q(f)(t)\right)^2 dt = \underset{s \in SN\ (q,\ \Delta\ (a,b))}{\text{Min}} \int_a^b \left(D^q(s)(t) - D^q(f)(t)\right)^2 dt$$

et tout autre élément de $SN\ (q, \Delta\ (a,b))$ ayant cette propriété diffère de s_i par un polynôme de degré q-1.

2°- $\forall\ s \in SN\ (q, \Delta\ (a,b))$

$$\int_a^b \left(D^q(s_i)(t) - D^q(s)(t)\right)^2 dt = \underset{f \in I_y}{\text{Min}} \int_a^b \left(D^q(f)(t) - D^q(s)(t)\right)^2 dt$$

et s_i est l'unique élément de $SN\ (q, \Delta\ (a,b))$ ayant cette propriété.

lemme 2

Soient : a un semi-produit scalaire sur un espace vectoriel X, P son noyau, p la semi-norme associée et V et W deux sous-espaces vectoriels de X tels que

$$a\ (V,W) = 0 \qquad \forall\ v \in V\ ,\ \forall\ w \in W$$

Soient V_τ = translaté de V

W_τ = translaté de W

et supposons que $\sigma \in V_\tau \cap W_\tau$ alors

1/ $\forall\ w \in W$

$$p\ (\sigma - w) = \operatorname*{Min}_{v \in V_\tau}\ p(v - w)$$

2/ $\forall\ v \in V$

$$p\ (\sigma - v) = \operatorname*{Min}_{w \in W_\tau}\ p\ (w - v)$$

(démonstration dans P.J. LAURENT / 7 /).

Démonstration du théorème 6

On applique le lemme 2 en posant :

$X = H^q\ (a,b)$

$$a\ (f,g) = \int_a^b D^q(f)\ D^q(g)\ (t)\ dt$$

$V = V_\tau = SN\ (q,\ \Delta\ (a,b))$

$$W = I_o = \{ f \in H^q(a,b) \ / \ f(t_i) = 0 \ ; \quad i = 1, 2 \ldots K \}$$

$$W = I_y = \{ f \in H^q(a,b) \ / \ f(t_i) = y_i \ ; \ i = 1,2 \ldots K \}$$

d'après le lemme 1 :

$$\int_a^b D^q(s)(t) \ D^q(f)(t) \ dt = 0$$

$$\forall \ s \in SN \ (q, \Delta \ (a,b)) \ ; \quad \forall \ f \in I_o$$

or

$$s_i \in SN \ (q, \Delta \ (a,b)) \qquad I_y = V \cap W$$

il suffit alors de prendre comme semi-norme

$$p \ (f) = \left[\int_a^b \left(D^q(f)(t) \right)^2 \ dt \right]^{1/2}$$

Corollaire

Soit $((t_1,y_1) \ (t_2,y_2) \ \ldots \ (t_K,y_K)) \in (\mathbb{R}^2)^K$ fixé

$(a = t_o < t_1 \ \ldots \ < t_K < t_{K+1} = b)$ et $q \in \mathbb{N}$;

et soit s_i l'unique élément de SN $(q, \Delta \ (a,b))$

(où $\Delta \ (a,b) = \{t_i\}_{i=1,K}$) tel que

$$s_i \ (t_j) = y_j \qquad j = 1,2 \ldots K$$

Alors

$$\int_a^b (D^q(s_i)(t))^2 \ dt = \operatorname*{Min}_{f \in I_y} \int_a^b (D^q \ (f)(t))^2 \ dt$$

et s_i est l'unique élément de I_y ayant cette propriété.

Démonstration

Il suffit de poser s ≡ o dans la seconde partie du théorème 6.

Remarque

Nous venons ainsi de caractériser le pouvoir lissant des fonctions-spline naturelles d'interpolation. En particulier, lorsqu'il s'agit de fonctions-spline cubiques (q = 2) nous vérifions bien qu'elles minimisent l'énergie de flexion (cf. § II.2).

II.8 - LES FONCTIONS-SPLINE D'AJUSTEMENT

Les fonctions-spline d'ajustement, sont, quant à elles, construites non pas pour prendre exactement des valeurs imposées, mais pour établir un compromis entre le lissage de la courbe et l'approximation des valeurs (ceci étant particulièrement intéressant, comme nous le verrons par la suite, pour approximer des données "bruitées").

Les démonstrations des résultats énoncés dans ce paragraphe sont analogues à celles développées pour les fonctions-spline d'interpolation (cf. / 7 /).

Théorème 7

Soit $y = (y_1, y_2 \dots y_K) \in \mathbb{R}^K$ un vecteur de données d'abscisses $(t_1, t_2 \dots t_K)$

avec $a = t_o < t_1 < t_2 \dots < t_K < t_{K+1} = b$

et soit $q \in \mathbb{N}$ fixé.

Alors il existe une et seule fonction-spline naturelle d'ordre q basée sur les noeuds $(t_1, t_2 \dots t_K)$ qui approxime les données $(y_1, y_2 \dots y_K)$ dans le sens suivant :

$$\forall\, (y_1\ y_2 \dots y_K) \in \mathbb{R}^K \quad \exists!\ s_a \in SN\ (q, \Delta\ (a,b))\ \text{telle que :}$$

$$(\text{où } \Delta\ (a,b) = \left\{t_i\right\}_{i=1,K})$$

$$s_a\ (t_i) + \frac{(-1)^q}{\rho}\ D^{2q-1}\ (s_a)(t_i^+) - D^{2q-1}\ (s_a)(t_i^-) = y_i$$

$$\rho \in \mathbb{R}_+^*$$

Définition 6 - Fonction-spline d'ajustement

L'unique solution s_a du problème

$((t_1,y_1)\ (t_2,y_2) \dots (t_K,y_K)) \in (\mathbb{R}^2)^K$ fixé, $q \in \mathbb{N}$ fixé

$$a < t_i < t_{i+1} < b \qquad i = 1,\ 2 \dots K-1$$

$$s(t_i) + \frac{(-1)^q}{\rho}\ \left[\ D^{2q-1}(s)(t_i^+) - D^{2q-1}(s)(t_i^-)\right] = y_i \qquad i = 1,\ 2 \dots K$$

$$s \in SN\ (q,\ \Delta\ (a,b)) \qquad \text{où } \Delta\ (a,b) = \{t_i\}_{i=1,K}$$

est appelée fonction-spline d'ajustement (d'ordre q) de paramètre ρ des données $(y_1\ y_2 \dots y_K)$.

Propriétés extrémales - Pouvoir lissant des fonctions-spline d'ajustement

Théorème 8

Soit $((t_1,y_1)\ (t_2,y_2)\ \ldots\ (t_K,y_K)) \in (\mathbb{R}^2)^K$ fixé

$(a = t_o < t_1 < \ldots < t_K < t_{KH} = b)$ et $q \in \mathbb{N}$

et soit s_a l'unique élément de $SN\ (q,\ \Delta\ (a,b))$

(où $\Delta\ (a,b) = \{t_i\}_{i=1,K}$) tel que :

$$s_a(t_i) + \frac{(-1)^q}{\rho}\left(D^{2q-1}(s_a)(t_i^+) - D^{2q-1}(s_a)(t_i)\right) = y_i$$

Alors :

$$\int_a^b \left(D^q(s_a)(t)\right)^2 dt + \rho \sum_{i=1}^{K} \left(s_a\ (t_i) - y_i\right)^2$$

$$= \min_{f \in H^q(a,b)} \int_a^b \left(D^q(f)(t)\right)^2 dt + \rho \sum_{i=1}^{K} \left(f(t_i) - y_i\right)^2$$

et s_a est l'unique élément de $H^q(a,b)$ ayant cette propriété.

III - LE DEBRUITAGE DES SPECTRES

III.1 - MÉTHODE DE DÉBRUITAGE DANS LE CAS D'UN BRUIT GAUSSIEN STATIONNAIRE

III.1.a - L'emploi des fonctions-spline d'ajustement

Débruiter un spectre consiste à trouver une courbe "suffisamment lisse" qui rende, autant que possible, toute l'information contenue dans les valeurs expérimentales ; et laisse de côté les oscillations dues à des perturbations extérieures au phénomène physique.

Soit donc :

$$y_i = f(t_i) + \varepsilon_i \qquad i = 1, 2 \ldots K$$

la forme des points du spectre

avec :

- $f(t_i)$: information cherchée
- ε_i : bruit au point d'abscisse $t_i \sim \mathcal{N}(0, \sigma_2)$ dans le cas d'un bruit gaussien stationnaire
- y_i : valeur mesurée expérimentalement au point d'abscisse t_i.

Débruiter le spectre revient à chercher une représentation correcte de f. Cette représentation devra satisfaire à la fois des conditions de régularité et de fidélité aux données

expérimentales ; ce qui nous conduit à considérer le problème de minimisation suivant : ($q \in \mathbb{N}$)

$$\min_{g \in H^q(a,b)} \int_a^b (D^q(g)(t))\, dt + \rho \sum_{i=1}^{K} (g(t_i) - y_i)^2$$

Et nous avons vu au paragraphe précédent (§ II.8) que ce problème admet comme solution la fonction-spline d'ajustement de paramètre ρ (fonction-spline que nous noterons s_ρ).

Remarque

Dans la pratique nous prendre q = 2 car les essais effectués en prenant q = 3 se sont avérés plus longs en temps-calcul pour des résultats analogues.

Il reste alors à donner une valeur judicieuse au paramètre ρ.

Le choix de ρ est fondamental pour les raisons suivantes :

- si ρ est trop grand, on favorise le deuxième terme de la fonctionnelle à minimiser et le résultat est une fonction-spline qui suit de trop près les données, c'est-à-dire qui ne "débruite" pas assez ;
- par contre, si ρ est trop petit, le phénomène inverse se produit ; la minimisation du terme correspondant au lissage est prépondérante et l'on risque d'éliminer des variations du spectre contenant une information physique.

En effet, soit σ^2 la variance du bruit.

Un débruitage correct est donné par :

$$\frac{1}{K} \sum_{i=1}^{K} \left(s_\rho (t_i) - y_i \right)^2 \cong \sigma^2$$

pour K assez grand

et pour un ρ trop grand on a :

$$\sum_{i=1}^{K} (g(t_i) - y_i)^2 \to 0$$

tandis que pour un ϱ trop petit :

$$\sum_{i=1}^{K} (g(t_i) - y_i)^2 \quad \text{devient trop grand.}$$

Pour déterminer le ρ optimal par rapport au problème posé, G. WAHBA / 23 / et P. CRAVEN et G. WAHBA / 4 / ont proposé l'emploi de la méthode "GENERALIZED CROSS VALIDATION". On trouvera dans la thèse de F. UTRERAS / 17 / la comparaison de cette méthode de choix de ρ avec d'autres déjà existantes.

III.1.b - Choix du paramètre d'ajustement - La Méthode de Validation Croisée

Nous nous situons dans le cas où les ε_i sont les réalisations de variables aléatoires qui suivent une loi de Gauss d'espérance nulle de variance σ^2 (i.e. $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ $\forall$ i = 1, 2 ... K).

La densité de probabilité d'un ε_i est :

$$\frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{t_i^2}{2\sigma^2}}$$

L'idée de la méthode de validation croisée est la suivante (STONE / 15 /) :

Soit $Y = \{y_i\}_{i=1,K}$ le vecteur des valeurs expérimentales du spectre et soit y_m l'une de ces valeurs.

Posons $\hat{Y}_m = Y \setminus \{y_m\}$

on a : $\hat{Y}_m \in \mathbb{R}^{K-1}$

Considérons la spline d'ajustement de $\hat{Y}_m$

$$S_\rho^{<m>} = \underset{g \in H^2(a,b)}{\text{Min}} \int_a^b \left(D^2(g)(t) \right)^2 dt + \rho \sum_{\substack{i=1 \\ i \neq m}}^{K} \left(g(t_i) - y_i \right)^2$$

que nous noterons aussi, en posant $\rho = \frac{\lambda}{K}$

$$(*) \quad S_\lambda^{<m>} = \underset{g \in H^q(a,b)}{\text{Min}} \int_a^b \left(D^2(g)(t) \right)^2 dt + \frac{\lambda}{K} \sum_{\substack{i=1 \\ i \neq m}}^{K} \left(g(t_i) - y_i \right)^2$$

(cette notation a l'avantage de mettre en évidence le terme :

$$\frac{1}{K} \sum \left(g(t_i) - y_i \right)^2$$

dont nous avons vu l'importance précédemment).

Supposons alors que λ_m soit un "bon choix" pour le paramètre λ en ce qui concerne le problème de minimisation (*) ; on devrait alors trouver que $S_\lambda^{<m>}(t_m)$ est un "bon estimateur" de la variable $f(t_m)$.

Il est donc naturel de choisir λ comme le réel qui vérifie le minimum de la fonction

$$V_o(\lambda) = \frac{1}{K} \sum_{m=1}^{K} \left(y_m - S_\lambda^{<m>}(t_m) \right)^2$$

Mais des arguments de convergence et de biais de l'estimation, développés par G. WAHBA / 4 / entre autres, ont poussé leur auteur à choisir de pondérer chaque terme de cette fonction. Ce qui se traduit par la recherche du minimisateur de la fonction :

$$V(\lambda) = \frac{1}{K} \sum_{m=1}^{K} \left\{ W_m(\lambda) \left[S^{<m>}(t_m) - y_m \right]^2 \right\}$$

Le choix des poids $W_m(\lambda)$ est motivé par le fait que pour K assez grand on souhaite avoir :

$$E(V(\lambda)) - \sigma^2 \approx E(R(\lambda))$$

où $$R(\lambda) = \frac{1}{K} \sum_{i=1}^{K} S_\lambda(t_i) - f(t_i)^2$$

Soit alors A_λ la matrice telle que :

$$(S_\lambda(t_i))_{i=1,K} = A_\lambda (y_i)_{i=1,K}$$

On choisit :

$$W_m(\lambda) = \frac{1 - a_{mm}(\lambda)}{\frac{1}{K} Tr(I - A_\lambda)}^2$$

où $a_{mm}(\lambda)$ désigne le $m^{ème}$ terme diagonal de la matrice A .

(cf. P. CRAVEN et G. WAHBA / 4 / et F. UTRERAS / 20 /).

Ce qui donne à la fonction de validation croisée V (λ) la valeur suivante :

$$V(\lambda) = \frac{1}{K} \frac{\sum_{i=1}^{K} \left(S_\lambda(t_i) - y_i \right)^2}{1 - \frac{1}{K} Tr(A_\lambda)}^2$$

Et, par conséquent, la méthode de validation croisée propose de choisir λ tel que :

$$V(\lambda) = \min_{\mu \in \mathbb{R}} V(\mu)$$

III.1.c - Le calcul du paramètre d'ajustement - L'algorithme d'UTRERAS

1°- Le problème de la trace de la matrice A_λ

Le calcul explicite de la fonction V (λ) et éventuellement de sa dérivée V'(λ) étant assez onéreux du fait de la nécessité de la connaissance de la matrice A_λ ; F. UTRERAS / 17 / a donné un algorithme qui détermine directement les valeurs propres de A_λ (ce qui est suffisant puis A_λ n'intervient que sous la forme de sa trace).

Soit $\Omega = (\omega_{ij})$ la matrice carrée d'ordre K telle que :

$$\omega_{ij} = D^3 (\gamma_i) (t_j^+) - D^3 (\gamma_i) (t_j^-)$$

où $\{ \gamma_1, \gamma_2, \ldots \gamma_K \}$ est la base canonique de SN (2, Δ (a,b))

$$a = t_o < t_1 \ldots < t_K < t_{K+1} = b$$

$$\Delta (a,b) = \{t_i\}_{i=1,K}$$

c'est-à-dire que

$$\gamma_i (t_j) = \delta_{ij}$$

Proposition

On peut alors écrire A_λ sous la forme suivante :

$$\boxed{A_\lambda = \left[I + \frac{K}{\lambda} \Omega \right]^{-1}}$$

(pour démonstration cf. F. UTRERAS / 19 /).

Calcul des valeurs propres de A_λ

Le calcul des valeurs propres de A_λ est désormais ramené à celui des valeurs propres de Ω.

Posons $t_i - t_{i-1} = h$

et

$$\Omega = \frac{1}{(Kh)^4} \Omega_K$$

alors :

$$Tr \left(I + \frac{K}{\lambda} \Omega \right)^{-1} = Tr \left(I + \frac{K}{\lambda (Kh)^4} \Omega_K \right)^{-1}$$

Soient $(\xi_1^K \ \xi_2^K \ \ldots \ \xi_K^K)$ les valeurs propre de Ω_K

$$Tr \left(I + \frac{K}{\lambda} \Omega \right)^{-1} = \sum_{i=1}^{K} \frac{1}{1 + \frac{K}{\lambda (hK)^4} \xi_i^K}$$

Notons :

$$B_K (\lambda) = \left(I + \frac{K}{\lambda} \Omega_K \right)^{-1}$$

alors :

$$Tr (A_\lambda) = Tr \left(I + \frac{K}{\lambda} \Omega \right)^{-1} = Tr (B_K (\lambda (Kh)^4))$$

et F. UTRERAS (cf. / 19 / démontre que si on note V_i $i = 1, 2 \ldots$ les valeurs propres du problème :

$$D^{(4)} (\phi) = V \phi$$

$$D^{(j)} (\phi) (0) = 0$$

$$D^{(j)} (\phi) (1) = 0 \qquad j = 2, 3$$

On a :

$$Tr\ (B_K\ (\lambda)) = \sum_{i=1}^{K} \frac{1}{1 + \frac{V_i}{\lambda}} + O\ (K^{-1})$$

et donc, finalement

$$T_r\ (A_\lambda) \cong \sum_{i=1}^{K} \frac{1}{1 + \frac{V_i}{\lambda\ (Kh)^4}}$$

2°- Le calcul du numérateur de V (λ)

Le calcul du numérateur de V (λ) nécessite la construction d'une spline S_λ pour un λ donné. Pour ce faire, on utilise un algorithme développé par L. PAIHUA / 9 / et basé sur une idée décrite par P.J. LAURENT / 7 / et qui est la "Mathode du Transport des relations pour les fonctions-spline d'ajustement". Le schéma de cette méthode est le suivant :

Soit W(t) le vecteur des dérivées successives de la fonction-spline d'ajustement d'ordre q s(t) :

$$W(t) = (D^i\ (s)\ (t))_{i=0\ \ldots\ 2q-1} \in \mathbb{R}^{2q}$$

On remarque que la connaissance de $W(t_i^+)$ ou de $W(t_{i+1}^-)$ détermine parfaitement le polynôme

$$p_i\ (t) = s\ (t)\ .\ \chi_{[t_i,\ t_{i+1}[}\ (t)$$

a) Supposons que l'on connaisse q relations linéaires indépendantes en t_i^- :

$$q \left\updownarrow \underbrace{\left(\mathcal{M}(t_i^-)\right)}_{2q} \quad \underbrace{\left(W(t_i^-)\right)}_{1} \right\updownarrow 2q = \underbrace{\left(\mu(t_i^-)\right)}_{1} \updownarrow q$$

Si on écrit la relation caractéristique des fonctions-spline d'ajustement

$$D^{2q-1}(s)(t_i^-) = D^{2q-1}(s)(t_i^+) + (-1)^q \quad s(t_i) - (-1)^q \quad y_i$$

il est alors possible de "transporter" ces q relations linéaires en t_i^+ :

$$(*) \qquad q \left\updownarrow \underbrace{\left(\mathcal{M}(t_i^+)\right)}_{2q} \quad \underbrace{\left(W(t_i^+)\right)}_{1} \right\updownarrow 2q = \underbrace{\left(\mu(t_i^+)\right)}_{1} \updownarrow q$$

avec pour $1 = 1, 2 \ldots q$

$$m_{1_j}(t_i^+) = \begin{cases} m_{1_j}(t_i^-) & \text{si } j = 2 \ldots 2q \\ m_{1_1}(t_i^-) + (-1)^q \rho \mu_{1,2q}(t_i^{-1}) & \text{sinon} \end{cases}$$

et

$$\mu_1(t_i^+) = \mu_1(t_i^-) + (-1)^q \rho y_i m_{1,2q}(t_i^-)$$

b) Supposons que l'on connaisse q relations linéaires indépendantes en t_i^+

On sait que s est un polynôme de degré 2q-1 dans l'intervalle t_i, t_{i+1}, donc on peut écrire :

$$2q \left\updownarrow \underbrace{\left(W\ (t_{i+1}^-) \right)}_{1} \right. = {}^{2q}\left\updownarrow \underbrace{\left(\mathcal{A}(h) \right)}_{2q} \right. \underbrace{\left(W\ (t_i^+) \right)}_{1} \left\updownarrow \right. 2q$$

avec $h = t_{i+1} - t_i$

$$a_{ij} = \begin{cases} h/(j-i)\ ! & \text{si } j > i \\ 0 & \text{sinon} \end{cases}$$

et de même

$$(**) \qquad 2q \left\updownarrow \underbrace{\left(W\ (t_i^+) \right)}_{1} \right. = \underbrace{\left(\mathcal{A}(-h) \right)}_{2q} \left\updownarrow \right. 2q \quad \underbrace{\left(W\ (t_{i+1}^-) \right)}_{1} \left\updownarrow \right. 2q$$

et donc en substituant (**) dans (*) on obtient :

$$\underline{\mathcal{M}(t_i^+)\ \mathcal{A}(-h)}_{\mathcal{M}(t_{i+1}^-)}\ W\ (t_{i+1}^-) = \underline{\mu\ (t_i^+)}_{\mu\ (t_{i+1}^-)}$$

Principe du transport des relations

On part de t_1^- où on connaît q relations linéaires (on sait que la fonction-spline est un polynôme de degré q-1 sur le premier intervalle)

$$D^j\ (s)\ (t_1^-) = 0 \qquad j = q, q+1, \ldots 2q-1$$

Ces relations peuvent être mises sous la forme :

$$(t_1^-)\ W\ (t_1^-) = \mu\ (t_1^-)$$

On applique alors la phase a) pour passer en t_1^+, puis la phase b) pour passer en t_2^-, et ainsi de suite jusqu'en t_K^+.

De la même façon, on part de t_K^+ où on connait aussi q relations linéaires

$$D^j(s)\ (t_K^+ = 0 \qquad\qquad j = q,\ qH,\ \ldots\ 2q-1$$

que l'on peut à nouveau mettre sous la forme :

$$(t_K^+)\ W\ (t_K^+) = \mu\ (t_K^+)$$

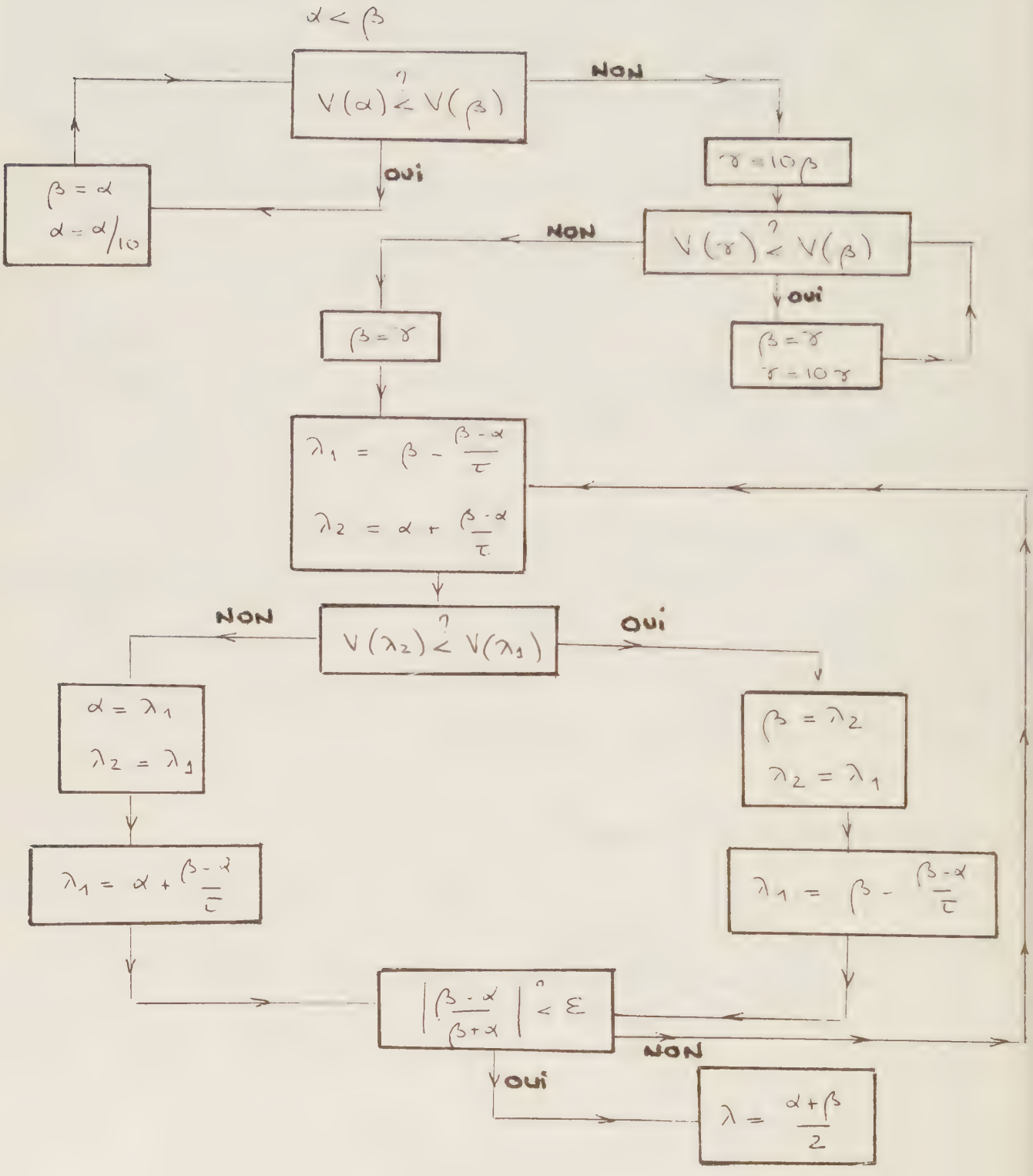
et on applique la phase a) pour passer en t_K^-, puis la phase b) (avec un pas (-h) pour passer en t_{K-1}^+), et ainsi de suite jusqu'en t_1^-.

Ces transports de relations permettent d'obtenir en chaque t_i^- (ou t_i^+) (pour i = 1, 2 ... K) q relations linéaires quand on passe de t_1 à t_K et q autres relations linéaires quand on va de t_K à t_1. Ces 2q relations linéaires (indépendantes) permettent de déterminer les 2q composantes du vecteur W (t_i^-) (ou W (t_i^+)) des dérivées de la spline s en t_i.

3°- La recherche du minimum de V (λ)

La recherche proprement dite, du réel qui minimise V (λ) est faite par la méthode de "section dorée" (cf. J. CEA / 3 /) que l'on peut résumer par l'organigramme suivant :

$$\tau = \frac{1 + \sqrt{5}}{2}$$


α < β
V(α) ? < V(β)
NON
OUI
γ = 10β
β = α
α = α/10
V(γ) ? < V(β)
NON
OUI
β = γ
β = γ
γ = 10γ
λ1 = β - (β - α)/τ
λ2 = α + (β - α)/τ
V(λ2) ? < V(λ1)
NON
OUI
α = λ1
λ2 = λ1
β = λ2
λ2 = λ1
λ1 = α + (β - α)/τ
λ1 = β - (β - α)/τ
|(β - α)/(β + α)| ? < ε
NON
OUI
λ = (α + β)/2

III.2 - MODIFICATION DE LA MÉTHODE DE DÉBRUITAGE DANS LE CAS D'UN BRUIT GAUSSIEN NON STATIONNAIRE

III.2.a - Pondération de l'ajustement

Les données expérimentales auxquelles nous sommes confrontés ici ont la forme suivante :

$$y_i = f(t_i) + \varepsilon_i \qquad i = 1, 2 \ldots K$$

où les $f(t_i)$ représentent l'information cherchée au point d'abscisse t_i et où les ε_i sont des réalisations de variables aléatoires qui suivent une loi de Gauss d'espérance nulle et de variance σ_i^2 (avec $\sigma_i^2 = f(ti)$

Pour tenir compte du fait que le "bruit" dépend du point que l'on considère, on choisit de lisser les données y_i, non plus par une fonction-spline d'ajustement définie au paragraphe II.8, mais par une fonction-spline d'ajustement pondéré.

Définition 7 - Fonction-spline d'ajustement pondéré

L'unique fonction S_{ρ,α_i} qui vérifie :

$$((t_1,y_1),(t_2,y_2) \ldots (t_K,y_K)) \in (\mathbb{R}^2)^K \text{ fixé }, \quad q \in \mathbb{N} \text{ fixé}$$

$$a = t_o < t_1 < t_2 < \ldots < t_K < t_{K+1} = b$$

$$\int_a^b \left(D^q (S_{\rho,\alpha_i})(t) \right)^2 dt + \rho \sum_{i=1}^{K} \alpha_i^2 \left(y_i - S_{\rho,\alpha_i}(t_i) \right)^2$$

$$= \min_{g \in H^q(a,b)} \left[\int_a^b \left(D^q(g)(t) \right)^2 dt + \rho \sum_{i=1}^{K} \alpha_i^2 \left(y_i - g(t_i) \right)^2 \right]$$

est appelée fonction-spline d'ajustement pondéré de paramètre ρ et de poids $(\alpha_i^2)_{i=1,K}$.

Notation

On notera, en adoptant de nouveau la notation $\rho = \frac{\lambda}{K}$

$$S_{\lambda,\alpha_i^2} = \min_{g \in H^q(a,b)} \int_a^b \left(D^q g(t) \right)^2 dt + \frac{\lambda}{K} \sum_{i=1}^{K} \alpha_i^2 \left(y_i - g(t_i) \right)^2$$

Choix des poids

F. UTREPAS / 17 / suggère de prendre :

$$\alpha_i^2 = \frac{1}{\sigma_i^2} \qquad \frac{1}{\frac{1}{K} \sum_{i=1}^{K} \frac{1}{\sigma_i^2}}$$

Ces poids sont tels que :

$$\frac{1}{K} \sum_{i=1}^{K} \alpha_i^2 = 1$$

Conséquence du choix des poids

Tout revient alors à chercher le paramètre optimal $\tilde{\lambda}$ de la fonction spline qui ajusterait le vecteur de données $\tilde{Y} = (\tilde{y}_i)_{i = 1, K}$ tel que :

$$\tilde{y}_i = f(t_i) + \tilde{\varepsilon}_i \qquad i = 1, 2 \ldots K$$

où $\tilde{\varepsilon}_i \sim \mathcal{N}(o, \sigma^2)$

C'est-à-dire qu'on se ramène au cas du débruitage dans le cas stationnaire en considérant que les $\tilde{\varepsilon}_i$ sont les réalisations de variables aléatoires qui suivent une loi de Gauss de variance nulle et d'espérance constante σ^2 (où σ^2 est la barycentre des variances σ_i^2 affectées des poids

$$p_i = \frac{\alpha_i^2}{K} \quad)$$

III.2.b - Précautions à prendre

Si l'un des $p_i = \frac{\alpha_i^2}{K}$ a une valeur nettement supérieure aux autres, le barycentre se trouve très proche de la variance associée, et ce, au détriment des autres variances.

Supposons, par exemple, que

$$p_m >> p_i \qquad \forall \; i = 1,2 \ldots K, \; i \neq m$$

Alors, le $\tilde{\lambda}$ que l'on va calculer sera très proche du λ_m optimal du problème :

$$\underset{g \in H^q(a,b)}{\text{Min}} \int_a^b \left(D^q(g)(t) \right)^2 dt + \frac{\lambda_m}{K} \sum_{i=1}^{K} \left(\tilde{\tilde{y}}_i - g(t_i) \right)^2$$

où $\quad \tilde{\tilde{y}}_i = f(t_i) + \tilde{\tilde{\varepsilon}}_i \qquad i = 1, 2 \ldots K$

avec

$$\tilde{\tilde{\varepsilon}}_i \sim (o, \sigma_m^2) \qquad \forall \; i = 1,2 \ldots K$$

Or, $p_i = \frac{\alpha_i^2}{K}$, donc p_i est proportionnel à α_i^2 et comme α_i^2 est inversement proportionnel à σ_i^2 ; p_i est inversement proportionnel à σ_i^2. Mais, dans le cas qui nous concerne, $\sigma_i^2 = y_i$, donc aux valeurs faibles du spectre correspondront

des poids très importants et, si l'on n'y prend garde, le optimal que l'on calculera correspondra à un problème où tous les points seraient entâchés d'une faible erreur (celle correspondant aux faibles valeurs du spectre) et on obtiendra, non plus un lissage, mais une interpolation.

D'autre part, si les plus grandes valeurs des α_i^2 sont très grandes par rapport aux autres et sont très peu nombreuses, le résultat du calcul du λ optimal ne peut être que de mauvaise qualité car on est alors amené à faire une estimation, basée sur la loi des grands nombres, sur un très faible nombre de valeurs.

Pour ces raisons, et du fait qu'à ma connaissance, aucun théorème de statistiques n'a été démontré dans le cas d'un bruit non stationnaire, on maniera le débruitage dans ce cas là avec beaucoup de prudence.

Remarque sur la trace de A_λ

Etant donné que nous avons constaté par l'expérience la faible sensibilité de l'ajustement aux variations légères de λ (seul, son ordre de grandeur importe) ; nous avons remplacé le calcul de la trace de A_λ pour le problème pondéré par celui du calcul de la trace de A_λ pour le problème stationnaire.

En effet, le calcul, dans le cas pondéré, de la trace de A_λ, nécessiterait pour chaque vecteur de données la recherche des valeurs propres du problème variationnel associé (cf. page 42) ; ce qui réclamerait un temps de calcul exagéré pour une précision dont l'importance semble minime au vu des résultats obtenus.

IV - LA DECONVOLUTION INSTRUMENTALE

Le problème de la déconvolution instrumentale se pose lorsque l'on a :

Signal = information cherchée * fonction instrument

Notations

. $f(x_i)$: valeur du signal au point d'abscisse x_i

pour $i = 1, 2, \ldots, N_{EXP}$

. $g(x_i)$: valeur de la fonction instrument au point d'abscisse x_i

pour $i = 1, 2, \ldots, N_{INST}$

. $u(x_i)$: valeur (que l'on cherche) de l'information au point d'abscisse x_i

pour $i = 1, 2, \ldots, N_{INF}$

IV.1 - MODÉLISATION DE LA FONCTION INSTRUMENT

Soit $Sg \in SN(2, \Delta(0, N_{INST}))$

où $\Delta(0, N_{INST})$ désigne une subdivision de l'intervalle $[0, N_{INST}]$

$$\Delta(0, N_{INST}) = \{r_m\}_{m=1,KR}$$

avec

$$0 = r_o < r_1 < r_2 \quad \ldots \quad < r_{KR} < r_{KR+1} = N_{INST}$$

D'après la proposition fondamentale sur les fonctions-spline polynômiales (§ II.4), nous pouvons écrire :

$$Sg\ (t) = c_1 + c_2\ t + \sum_{m=1}^{KR} c_{2+m}\ \frac{(t - r_m)_+^3}{3\ !}$$

avec

$$\sum_{m=1}^{KR} c_{2+m} = 0$$

$$\sum_{m=1}^{KR} r_m\ c_{2+m} = 0$$

Calcul des paramètres $(c_m)_{m=1,KR+2}$ de Sg

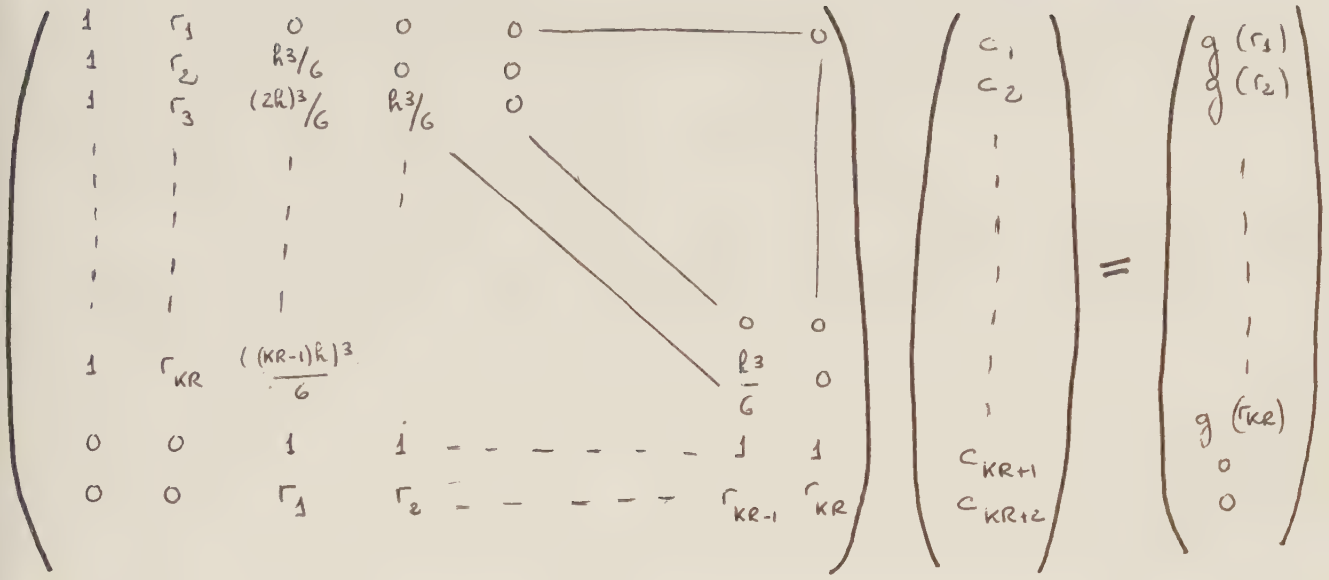
Pour calculer les KR+2 paramètres de la fonction-spline Sg qui modelise la fonction-instrument, nous allons résoudre le système linéaire de KR+2 équations à KR+2 inconnues suivant :

$$(P_1) \begin{cases} Sg\ (r_m) = g\ (r_m) & m = 1, 2, \ldots KR \\ \sum_{m=1}^{KR} c_{2+m} = 0 \\ \sum_{m=1}^{KR} r_m\ c_{2+m} = 0 \end{cases} \Bigg\} \text{contraintes imposées aux paramètres d'une fonction-spline polynômiale naturelle}$$

Supposons par exemple que les points de jonctions (r_m) sont également répartis dans l'intervalle $0, N_{INST}$ et que

$$r_m = mh$$

Le système (P_1) s'écrit alors sous forme matricielle :



Soit :

$$KR+2 \left\{ \begin{matrix} (A) \\ KR+2 \end{matrix} \right. \quad \left. \begin{matrix} (C) \\ 1 \end{matrix} \right\} KR+2 \quad = \quad \left. \begin{matrix} (G) \\ 1 \end{matrix} \right\} KR+2$$

Or la matrice A possède une seule surdiagonale et donc se triangularise très facilement.

Nous obtenons :

$$\boxed{C = A^{-1} G}$$

ce qui nous donne la valeur des paramètres $(c_i)_{i=1,KR+2}$ et nous détermine parfaitement la fonction-spline naturelle :
Sg SN (2, Δ (O, N_{INST})) qui modélise la fonction-instrument g.

IV.2 - MODÉLISATION DE L'INFORMATION CHERCHÉE

Nous choisissons de modéliser l'information cherchée par une fonction-spline polynômiale naturelle d'ordre 2. Soit S_u cette fonction-spline

$$S_u \in SN\ (2,\ \Delta\ (0,\ N_{INF}))$$

où $\Delta\ (0,\ N_{INF})$ désigne une subdivision de l'intervalle $0,\ N_{INF}$

$$\Delta\ (0,\ N_{INF}) = \{t_1\}_{1=1,KT}$$

avec :

$$0 = t_o < t_1 < t_2 \ldots < t_{KT} < t_{KT+1} = N_{INF}$$

D'après la proposition fondamentale (§ II.4), nous pouvons écrire :

$$S_u\ (t) = b_1 + b_2\ t + \sum_{1=1}^{KT} b_{2+1}\ \frac{(t - t_1)_+^3}{3\ !}$$

avec

$$\sum_{1=1}^{KT} b_{2+1} = 0$$

$$\sum_{1=1}^{KT} t_1\ b_{2+1} = 0$$

IV.3 - MODÉLISATION DU SIGNAL

Nous allons modéliser le signal (que nous connaissons), par le produit de convolution de la fonction-spline modélisant l'information cherchée par la fonction-spline modélisant la fonction instrument.

D'après les résultats du paragraphe II.5, nous pouvons écrire que sur le plus petit des deux supports de S_u et de S_g on a :

$$S_f(x) = (S_g * S_u)(x) = \int_o^x S_g(x-t)\, S_u(t)\, dt$$

c'est-à-dire que pour x appartenant au plus petit des deux supports de S_u et de S_g on a :

$$S_f(x) = b_1 c_1 x + b_1 c_2 x^2 + b_2 c_1 \frac{x^2}{2} + b_2 c_2 \frac{x^3}{3!}$$

$$+ b_1 \sum_{m=1}^{KR} c_{2+m} \frac{(x-r_m)_+^4}{4!}$$

$$+ b_2 \sum_{m=1}^{KR} c_{2+m} \frac{(x-r_m)_+^5}{5!}$$

$$+ c_1 \sum_{l=1}^{KT} b_{2+l} \frac{(x-t_l)_+^4}{4!}$$

$$+ c_2 \sum_{l=1}^{KT} b_{2+l} \frac{(x-t_l)_+^5}{5!}$$

$$+ \sum_{l=1}^{KT} \sum_{m=1}^{KR} b_{2+l}\, c_{2+m} \frac{(x - t_l - r_m)_+^7}{7!}$$

Remarque

S_f est une fonction linéaire des paramètres : $(b_1, b_2 \ldots b_{KT+2})$ de la fonction-spline S_u.

IV.4 - FORMULATION DU PROBLÈME ET RÉSOLUTION

Nous cherchons à déterminer les KT+2 paramètres de la fonction-spline S_u. Pour ce faire, nous devons résoudre le système d'équations linéaires suivant :

$$(P_2) \quad \begin{cases} S_f(x_i) = f(x_i) & \text{pour } i = 1, \text{Inf}(N_{INF}, N_{INST}) \\ \left.\begin{array}{l} \sum_{l=1}^{KT} b_{2+l} = 0 \\ \sum_{l=1}^{KT} t_l \, b_{2+l} = 0 \end{array}\right\} & \text{contraintes imposées aux paramètres d'une fonction-spline polynômiale naturelle} \end{cases}$$

IV.4.1 - Problème de l'ajustement à l'origine

La fonction-spline cubique S_u est formée d'un morceau de polynôme de degré 1 sur le premier intervalle $[o, t_1[$; ce qui, comme le signalent M.J.C. POWELL / 10 / et C. DE BOOR / 2 /, manque de souplesse pour modéliser le départ de la courbe de l'information cherchée.

Pour pallier cet inconvénient nous choisissons de construire une fonction artificielle $\tilde{u}$ telle que :

$$\begin{cases} \tilde{u}(x_i) = 0 & \text{pour } i = 1, IA \\ \tilde{u}(x_i) = u(x_{i-IA+1}) & \text{pour } i = IA+1, N_{INF} + IA \end{cases}$$

Et de la même façon, nous prenons $\tilde{g}$ telle que :

$$\begin{cases} \tilde{g}(x_i) = 0 & \text{pour } i = 1, IB \\ \tilde{g}(x_i) = g(x_{i-IB+1}) & \text{pour } i = IB+1, N_{INST} + IB \end{cases}$$

De cette façon, en choisissant de prendre le premier point de jonction t_1 de la spline S_u tel que :

$$x_1 < t_1 < x_{IA}$$

nous évitons les problèmes dus au manque de souplesse à l'origine de la courbe.

En ce qui concerne la convolée f, on a alors à faire à une nouvelle fonction :

$$\tilde{f} = \tilde{u} * \tilde{g}$$

qui est telle que

$$\tilde{f}(x_i) = 0 \qquad \text{pour } i = 1, IA + IB$$

$$\tilde{f}(x_i) = f(x_{i-IA-IB+1}) \qquad \text{pour } i = IA + IB + 1, N_{EXP} + IA + IB$$

IV.4.b - Problème du support

Le système (P_2) n'est valable que sur le plus petit des deux supports de S_u et de S_g. Or, dans la réalité, la longueur du support de u (information cherchée) est de l'ordre de 20 fois celle du support de g (fonction-instrument).

Et comme f = u * g et que toutes ces fonctions sont à valeurs positives, on a :

$$\text{supp. } f = \text{supp. } u + \text{supp. } g$$

Donc la modélisation proposée pour f ne serait valable que sur 1/21ème de son support. C'est pour cette raison que nous allons artificiellement rallonger les supports de u et g de la façon suivante :

Nous construisons $\tilde{\tilde{u}}$ et $\tilde{\tilde{g}}$ telles que :

1
$$\begin{aligned} &\text{supp. } \tilde{\tilde{u}} = \text{supp. } \tilde{f} \\ &\text{supp. } \tilde{\tilde{g}} = \text{supp. } \tilde{f} \end{aligned}$$

2
$$\tilde{u} \equiv \begin{cases} \hat{u} & \text{sur supp. } \tilde{u} \\ 0 & \text{sur supp. } \tilde{f} \setminus \text{supp. } \tilde{u} \end{cases}$$

3
$$\tilde{\tilde{g}} \equiv \begin{cases} \tilde{g} & \text{sur supp. } \tilde{g} \\ 0 & \text{sur supp. } \tilde{f} \setminus \text{supp. } \tilde{u} \end{cases}$$

IV.4.c - Résolution

Soit N le nombre de valeurs prises par $\tilde{f}$.

$$N = N_{EXP} + IA + IB$$

- 1ère étape

Nous résolvons le système $(\tilde{P}_1)$ (analogue au système P_1 du paragraphe IV.2) :

$$S_g(r_m) = \tilde{\tilde{g}}(r_m) \qquad m = 1, 2 \ldots KR$$

$$(\tilde{P}_1) \quad \begin{cases} \sum_{m=1}^{KR} c_{2+m} = 0 \\ \sum_{m=1}^{KR} r_m \, c_{2+m} = 0 \end{cases}$$

et nous en déduisons les KR+2 paramètres $(c_i)_{i=1,KR+2}$ de la spline S_g.

- 2ème étape

Le système (P_2) (page 57) s'écrit maintenant pour la fonction $\tilde{f}$:

$$\tilde{(P_2)} \quad \begin{cases} S_f(x_i) = \tilde{f}(x_i) & \text{pour } i = 1,N \\ S_u(x_i) = 0 & \text{pour } i = 1, IA \\ S_u(x_i) = 0 & \text{pour } i = N_{INF} + IA+1, N \\ \sum_{l=1}^{KT} b_{2+l} = 0 & \\ \sum_{l=1}^{KT} t_l \, b_{2+l} = 0 & \end{cases}$$

V - L'AUTODECONVOLUTION

Le problème de l'autodéconvolution se pose lorsque l'on a :

Signal = information cherchée * information cherchée

Notations

. $f(x_i)$: valeur du signal au point d'abscisse x_i

pour $i = 1, 2, \ldots N_{EXP}$

. $u(x_i)$: valeur (que l'on cherche) de l'information au point d'abscisse x_i

pour $i = 1,2, \ldots N_{INF}$

VI.1 - MODÉLISATION DE L'INFORMATION CHERCHÉE

De la même façon que dans déconvolution instrumentale nous allons considérer une fonction-spline polynômiale naturelle d'ordre 2 que nous notons S_u :

$$S_u \in SN\left(2, \Delta\left(0, N_{INF}\right)\right)$$

où $\Delta(0, N_{INF})$ désigne une subdivision $\{t_i\}_{u=1,KT}$ de l'intervalle $[0, N_{INF}]$ telle que :

$$0 = t_o < t_1 \ldots < t_{KT} < t_{KT+1} + N_{INF}$$

D'après la proposition fondamentale (§ II.4) nous pouvons écrire :

$$S_u(t) = b_1 + b_2\, t + \sum_{l=1}^{KT} b_{2+1}\, \frac{(t - t_1)_+^3}{3\,!}$$

avec

$$\sum_{l=1}^{KT} b_{2+1} = 0$$

$$\sum_{l=1}^{KT} t_1\, b_{2+1} = 0$$

V.2 - MODÉLISATION DU SIGNAL

Nous allons de nouveau modéliser le signal (connu), par le produit d'autoconvolution de la fonction-spline qui modélise l'information cherchée.

D'après les résultats du paragraphe II.5, nous pouvons écrire que sur le rapport de S_u :

$$S_f(x) = (S_u * S_u)(x) = \int_o^x S_u(x-t)\, S_u(t)\, dt$$

c'est-à-dire que pour x appartenant au support de S_u on a :

$$S_f(x) = b_1^2\, x + 2\, b_1 b_2\, \frac{x^2}{2} + b_2^2\, \frac{x^3}{3!} + 2 \sum_{l=1}^{KT} b_1\, b_{2+1}\, \frac{(x-t_1)_+^4}{4\,!}$$

$$+ 2 \sum_{l=1}^{KT} b_2 \, b_{2+l} \, \frac{(x - t_l)_+^5}{5\,!}$$

$$+ \sum_{l=1}^{KT} \sum_{m=1}^{KT} b_{2+l} \, b_{2+m} \, \frac{(x - t_l - r_m)_+^7}{7\,!}$$

Remarque

S_f est une fonction non linéaire des paramètres $(b_1 \; b_2 \; \ldots\ldots \; b_{KT+2})$ de la fonction-spline S_u.

V.3 - FORMULATION DU PROBLÈME ET RÉSOLUTION

Nous cherchons, cette fois encore, à déterminer les KT + 2 paramètres de la spline S_u. Nous sommes donc conduits à résoudre le système d'équations non linéaires suivant :

$$(P_3) \quad \begin{cases} S_f(x_i) = f(x_i) & \text{pour } i = 1, N_{INF} \\ \sum_{l=1}^{KT} b_{2+l} = 0 \\ \sum_{l=1}^{KT} t_l \, b_{2+l} = 0 \end{cases} \quad \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \begin{array}{l} \text{contraintes imposées} \\ \text{aux paramètres d'une} \\ \text{fonction-spline} \\ \text{polynômiale naturelle} \end{array}$$

V.3.a - Problème de l'ajustement à l'origine

Pour les mêmes motifs que ceux évoqués dans la déconvolution instrumentale (cf. § IV.3.a), nous choisissons de prendre une fonction $\tilde{u}$ telle que :

$$\tilde{u}(x_i) = 0 \qquad \text{pour } i = 1, IC$$

$$\tilde{u}(x_i) = u(x_{i-IC+1}) \qquad \text{pour } i = IC+1, N_{INF} + IC$$

Et donc, nous aurons une nouvelle autoconvolée $\tilde{f}$ telle que :

$$\tilde{f}(x_i) = 0 \qquad \text{pour } i = 1,\ IC + IC$$

$$\tilde{f}(x_i) = f(x_{i-2-IC+1}) \qquad \text{pour } i = IC + IC+1,\ N_{EXP} + IC + IC$$

V.3.b - Problème du support

Le système (P_3) n'est valable que sur le support de S_u ; nous sommes donc amenés à utiliser un artifice du même style que celui employé dans la méthode de déconvolution instrumentale (cf. § IV.4.b). Nous construisons une nouvelle fonction $\tilde{\tilde{u}}$ telle que :

1 $\quad \text{supp. } \tilde{\tilde{u}} = \text{supp. } \tilde{f}$

2 $$\tilde{\tilde{u}} \equiv \begin{cases} \tilde{u} & \text{sur supp. } \tilde{u} \\ o & \text{sur supp. } \tilde{f} \setminus \text{supp. } \tilde{u} \end{cases}$$

V.3.c - Résolution

Soit N le nombre de valeurs prises par $\tilde{f}$

$$N = N_{EXP} + IC + IC$$

Le système (P_3) (page 63) s'écrit maintenant pour la fonction $\tilde{f}$:

$$(\tilde{P}_3) \quad \begin{cases} S_f(x_i) = \tilde{f}(x_i) & \text{pour } i = 1, 2, \ldots N \\ S_u(x_i) = 0 & \text{pour } i = 1, IC \\ S_u(x_i) = 0 & \text{pour } i = N_{INF} + IC+1, N \\ \sum_{l=1}^{KT} b_{2+l} = 0 & \\ \sum_{l=1}^{KT} t_l \, b_{2+l} = 0 & \end{cases}$$

VI - REALISATION ET EXPLOITATION

VI.1 - MISE EN OEUVRE

VI.1.a - Mise en oeuvre du débruitage

Il convient, tout d'abord, de modifier la fonction $V(\lambda)$ donnée au paragraphe III.1.b pour l'adapter à la méthode d'ajustement pondéré.

Nous avions alors :

$$V(\lambda) = \frac{\frac{1}{K} \sum_{1=1}^{K} \left(y_i - S_\lambda (t_i) \right)^2}{\left(1 - \frac{1}{K} \mathrm{Tr}\,(A_\lambda) \right)^2}$$

Pour une fonction-spline d'ajustement pondéré, le paramètre de lissage optimal λ sera celui pour lequel la fonction $V(\lambda)$ est minimale, avec :

$$V(\lambda) = \frac{\frac{1}{K} \sum_{i=1}^{K} \alpha_i^2 \left(y_i - S_\lambda (t_i) \right)^2}{\left(1 - \frac{1}{K} \mathrm{Tr}\,(A_\lambda) \right)^2}$$

où

$$\alpha_i^2 = \frac{1}{y_i} \quad \frac{1}{\frac{1}{K} \sum_{i=1}^{K} \frac{1}{y_i}} \qquad (*)$$

Toutefois, pour éviter les problèmes (évoqués au paragraphe II.2.b), dus aux trop faibles valeurs du spectre, nous utiliserons les α_i^2 définis par la relation (*) en posant :

$$\frac{1}{y_i} = 0 \qquad \text{si} \qquad y_i < \frac{5 \times y_{max}}{100}$$

où y_{max} désigne la plus grande valeur du spectre.

VI.1.b - Mise en oeuvre de la déconvolution instrumentale

Nous avons donc à résoudre le système linéaire par rapport aux paramètres (de la fonction-spline S_u) que nous cherchons (cf. § IV.4.c) :

$$\tilde{(P_2)} \begin{cases} (1) \quad S_f(x_i) = \tilde{f}(x_i) & \text{pour } i = 1,N \\ (2) \quad S_u(x_i) = 0 & \text{pour } i = 1,IA \\ (3) \quad S_u(x_i) = 0 & \text{pour } i = N_{INF} + IA+1, N \\ (4) \quad \sum_{l=1}^{KT} b_{2+l} = 0 & \\ (5) \quad \sum_{l=1}^{KT} t_l \, b_{2+l} = 0 & \end{cases}$$

où :

. l'équation (1) représente l'égalité entre la convolée des fonctions-spline "modèles" S_u et S_g, et le spectre expérimental.

. les équations (2) et (3) représentent les contraintes imposées sur le départ et le support de S_u.

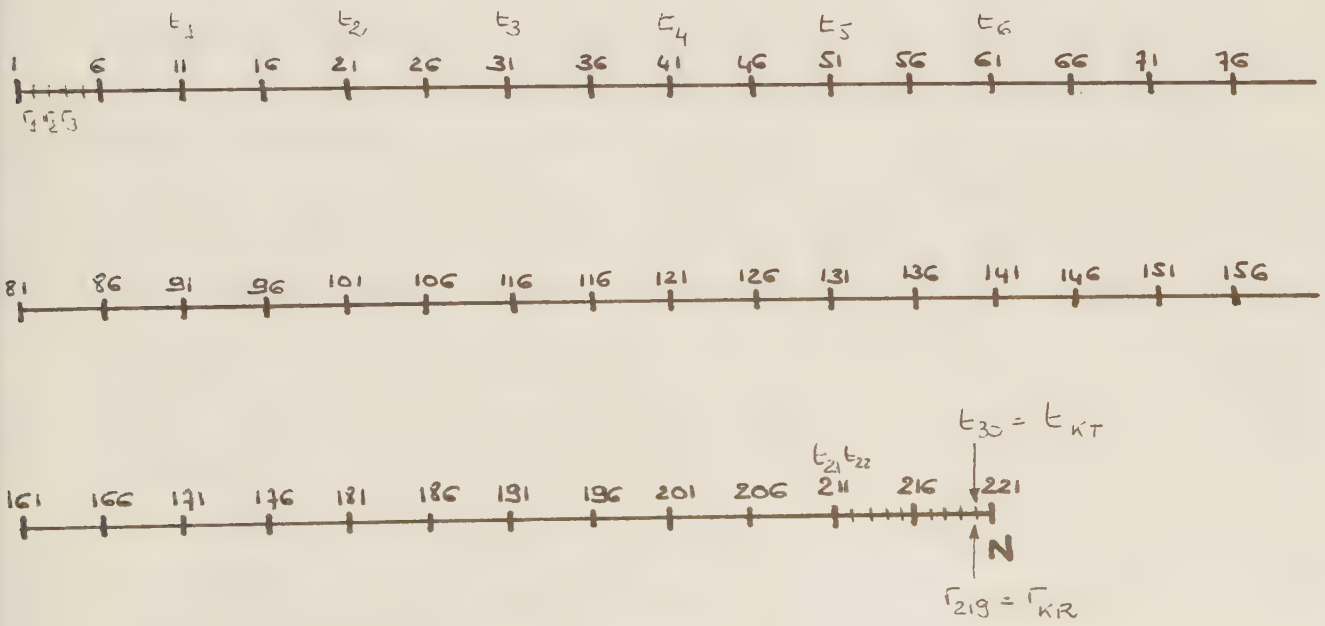
. les équations (4) et (5) représentent les contraintes imposées sur les paramètres de la fonction-spline naturelle S_u.

Remarque

Dans la pratique nous connaissons le signal f donné par N_{EXP} valeurs et la fonction instrument g donnée par N_{INST} valeurs. L'information cherchée u aura N_{INF} valeurs, avec :

$$N_{INF} + N_{INST} = N_{EXP}$$

Le choix des subdivisions



- Les subdivisions de S_g

$$r_m = mh \qquad m = 1, N-2$$

- Les subdivisions de S_u

$$t_1 = 10\ 1h \qquad \text{si } 1 = 1,\ N_{EXP}/10$$

$$t_1 = N_{EXP}/10\ \ 1h + (1-N_{EXP})h$$

$$\text{si } 1 = N_{EXP}/10 + 1,\ N_{EXP}/10 + 9$$

Ces choix de subdivisions sont motivés à la fois par le souhait de récolter le maximum d'informations et par le souci de ne pas avoir besoin de trop grandes places-mémoire lors des calculs.

Le choix de IA et de IB

IA et IB représentent respectivement le nombre de composantes nulles qui précèdent la première composante des vecteur $(g(x_i))_{i=1,N_{INST}}$ et $(u(x_i))_{i=1,N_{INF}}$.

Etant donné le choix du pas entre deux points de jonction t_i et t_{i+1}, que nous avons fait, à savoir :

$$t_{i+1} - t_i = 10\ h \qquad i = 1,\ N_{EXP}/10$$

nous prendrons $\boxed{IB = 10}$ de façon à avoir le premier point de jonction t_1 qui corresponde à l'ajustement sur une valeur nulle. De la sorte l'ajustement sur les premières valeurs de la fonction u se fera à l'aide de morceaux de polynômes de degré 3 (et non pas de degré 1).

Par analogie, nous choisissons aussi : $\boxed{IA = 10}$.

Le choix du pas de résolution

Ce choix, comme les précédents, est motivé, à la fois par les besoins pour posséder un maximum d'informations et par les restrictions imposées par les problèmes de place-mémoire et de temps-calcul.

Nous écrivons donc l'équation (1) sous la forme :

(1') $S_f(x_i) = x_i \qquad i = \alpha\ hpas$

avec $hpas = 5h$

$\alpha = 1, 2 \ldots$

L'exécution

Du fait des choix des subdivisions et de IB, les équations (2) et (3) deviennent respectivement :

(2') $S_u(t_1) = 0$

(3') $S_u(t_1) = 0 \qquad 1 = N_{EXP}/10 \ ; KT$

où $KT = N_{EXP}/10 + IB-1$

Le système { (1'), (2'), (3'), (4), (5) } est un système redondant d'équations linéaires des paramètres $(b_i)_{i=1,KT+2}$ de la fonction-spline polynômiale naturelle S_u qui modélise l'information cherchée u. Ce système s'écrit sous la forme matricielle :

$$AB = Z$$

où :

1°- $A = (a_{ij})$ $\quad i = 1,\ N/5 + IB+3$
$\quad = 1,\ KT+2$

est la matrice rectangle définie comme suit :

- pour i = 1, N/5

$$a_{i1} = c_1 x + c_2 \frac{x^2}{2} + \sum_{m=1}^{KR} c_{2m} \frac{(x - r_m)_+^4}{4!}$$

$$a_{i2} = c_1 \frac{x^2}{2} + c_2 \frac{x^3}{6} + \sum_{m=1}^{KR} c_{2m} \frac{(x - r_m)_+^5}{5!}$$

$j = 3,\ KT+2$

$$a_{ij} = c_1 \frac{(x - t_{j-2})_+^4}{4!} + c_2 \frac{(x - t_{j-2})_+^5}{5!} + \sum_{m=1}^{KR} c_{2+m} \frac{(x - t_{j-2} - r_m)_+^7}{7!}$$

où $x = i \times hpas$

- pour i N/5 + 1, N/5 + IB

$a_{i1} = 1$

$a_{i2} = x$

$j = 3,\ KT+2$

$$a_{ij} = \frac{(x - t_{j-2})_+^3}{3!}$$

où $x = N/5 \times hpas + (i - N/5)h$

- pour i = N/5 + IB+1

a_{i1} 1

$a_{i2} = t_1$

j = 3, KT+2

$a_{ij} = 0$

- pour i = N/5 + IB+2

$a_{i1} = 0$

$a_{i2} = 0$

j = 3, KT+2

$a_{ij} = 1$

- pour i = N/5 + IB+3

$a_{i1} = 0$

$a_{i2} = 0$

j = 3, KT+2

$a_{ij} = t_{j-2}$

2°- $Z = (Z_j)_{j=1,\ N/5 + IB+3}$

est le vecteur défini comme suit :

- pour j = 1, N/5

$$Z_j = f(x_j) \qquad \text{où} \quad x_j = j \times hpas$$

- pour j = N/5 +1, N/5 + IB+3

$$Z_j = 0$$

3°- $B = (b_j)_{j=1,\ KT+2}$

où les b_j sont les paramètres de la fonction-spline S_u.

Ce système linéaire surabondant est résolu par un sous-programme de la CISI, codé DMAL1 (cf. Annexe I).

VI.1.c - Mise en oeuvre de l'autoconvolution

Le problème est beaucoup plus délicat que le précédent parce que le système $(\tilde{P}_3)$ (cf. § V.3.c) n'est pas un système linéaire des paramètres $(b_i)_{i=1,KT+2}$ de la fonction-spline S_u.

$$(\tilde{P}_3) \begin{cases} (1) \quad S_f(x_i) = \tilde{f}(x_i) & \text{pour } i = 1,N \\ (2) \quad S_u(x_i) = 0 & \text{pour } i = 1,IC \\ (3) \quad S_u(x_i) = 0 & \text{pour } i = N_{INF} + IC+1,\ N \\ (4) \quad \sum_{l=1}^{KT} b_{2+l} = 0 & \\ (5) \quad \sum_{l=1}^{KT} t_l\, b_{2+l} = 0 & \end{cases}$$

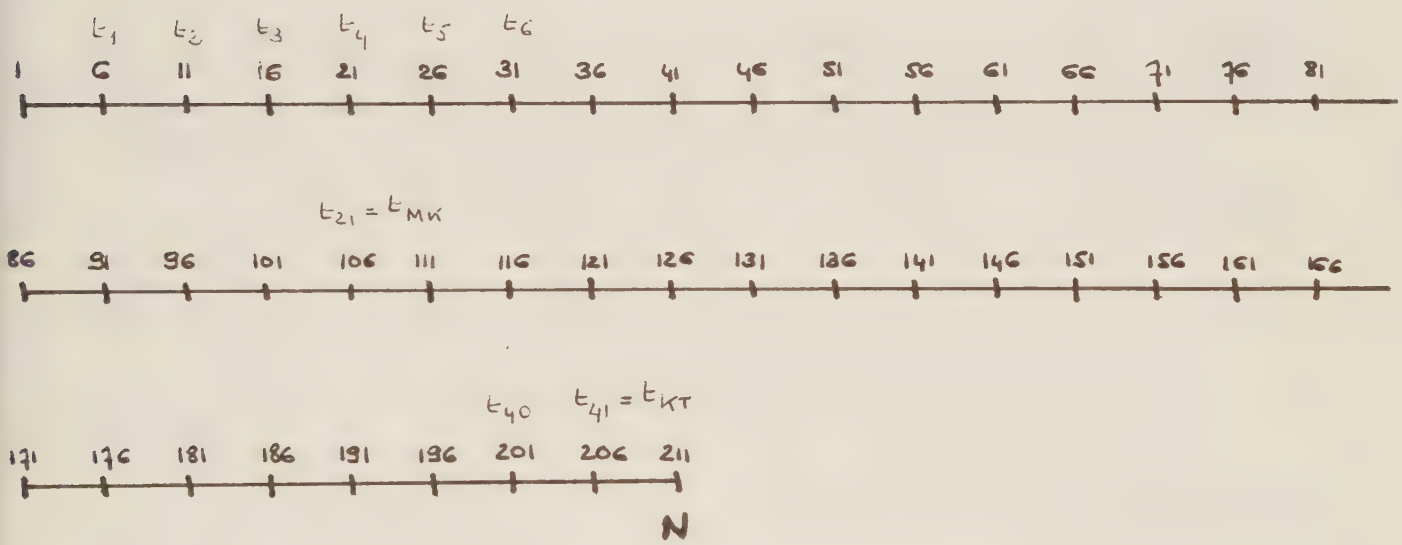
où :

. l'équation (1) représente l'égalité entre l'autoconvolée de la spline qui modélise l'information cherchée et le spectre expérimental.

. les équations (2) et (3) représentent les contraintes imposées sur le départ et sur le support de la fonction-spline S_u qui modélise l'information cherchée.

. les équations (4) et (5) représentent les contraintes imposées aux paramètres d'une fonction-spline polynômiale naturelle.

Le choix des subdivisions



Les subdivisions du S_u sont choisies de la façon suivante :

$$t_1 = 5\ 1h$$

Nous noterons :

$$MK = KT/2 + 1$$

Le choix de IC

IC représente le nombre de composantes nulles que l'on ajoute au vecteur $(u(x_i))_{i=1,N_{INF}}$ avant sa première composante propre.

Etant donné que nous avons choisi de prendre :

$$t_{i+1} - t_i = 5h \qquad i = 1, KT$$

nous décisons d'avoir $\boxed{IC = 5}$ de façon à ce que le premier point de jonction t_1 corresponde à l'ajustement sur une valeur nulle.

Le choix du pas de résolution

Toujours pour des raisons de place-mémoire, nous choisissons d'écrire l'équation (1) sous la forme :

(1') $$S_f(x_i) = \tilde{f}(x_i) \qquad i = hpas$$

avec $hpas = 5h$

$= 1, 2 \ldots$

L'exécution

Du fait du choix des subdivisions et de IC, les équations (2) et (3) s'écrivent :

(2') $$S_u(t_1) = 0$$

(3') $$S_u(t_1) = 0 \qquad 1 = MK, KT$$

Le système { (1'), (2'), (3'), (4), (5) } est un système d'équations non linéaires des paramètres $(b_i)_{i=1,KT+2}$ de la

fonction-spline polynômiale naturelle S_u qui modélise l'information cherchée.

Nous choisissons de résoudre un tel système non linéaire en résolvant par moindres carrés l'équations non linéaire (1') sans les contraintes linéaires (2'), (3'), (4) et (5). C'est-à-dire que nous cherchons le minimum suivant :

$$\min_{b=(b_i)\ Q^{KT+2}} \left(\sum_{i=1}^{N} \tilde{f}(x_i) - S_f(x_i, b) \right)^2$$

$$(4) \qquad \sum_{l=1}^{KT} b_{2+l} = 0$$

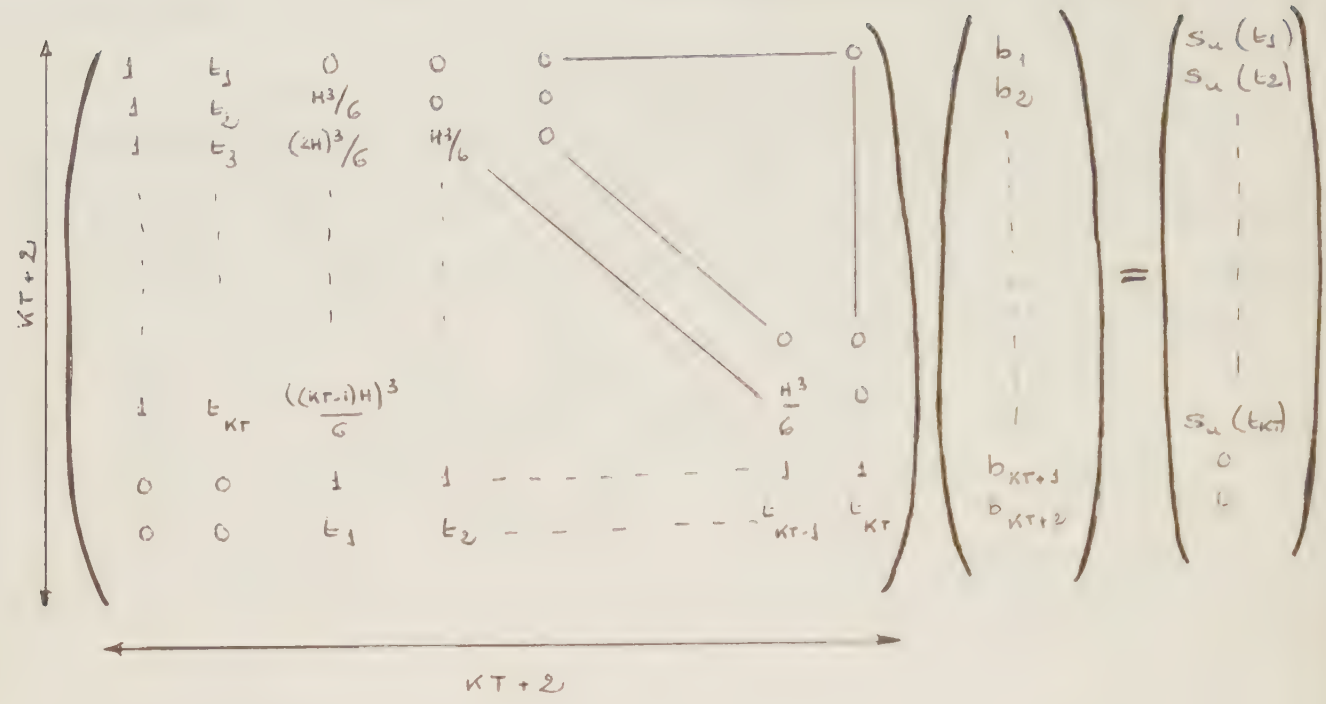
$$(5) \qquad \sum_{l=1}^{KT} t_l\, b_{2+l} = 0$$

$$(2') \qquad S_u(t_1) = 0$$

$$(3') \qquad S_u(t_l) = 0 \qquad l = MK, KT$$

Donnons maintenant la forme explicite des contraintes linéaires :

1°- Ecrivons tout d'abord la relation linéaire qui lie une fonction-spline naturelle d'ordre 2 à ses coefficients :

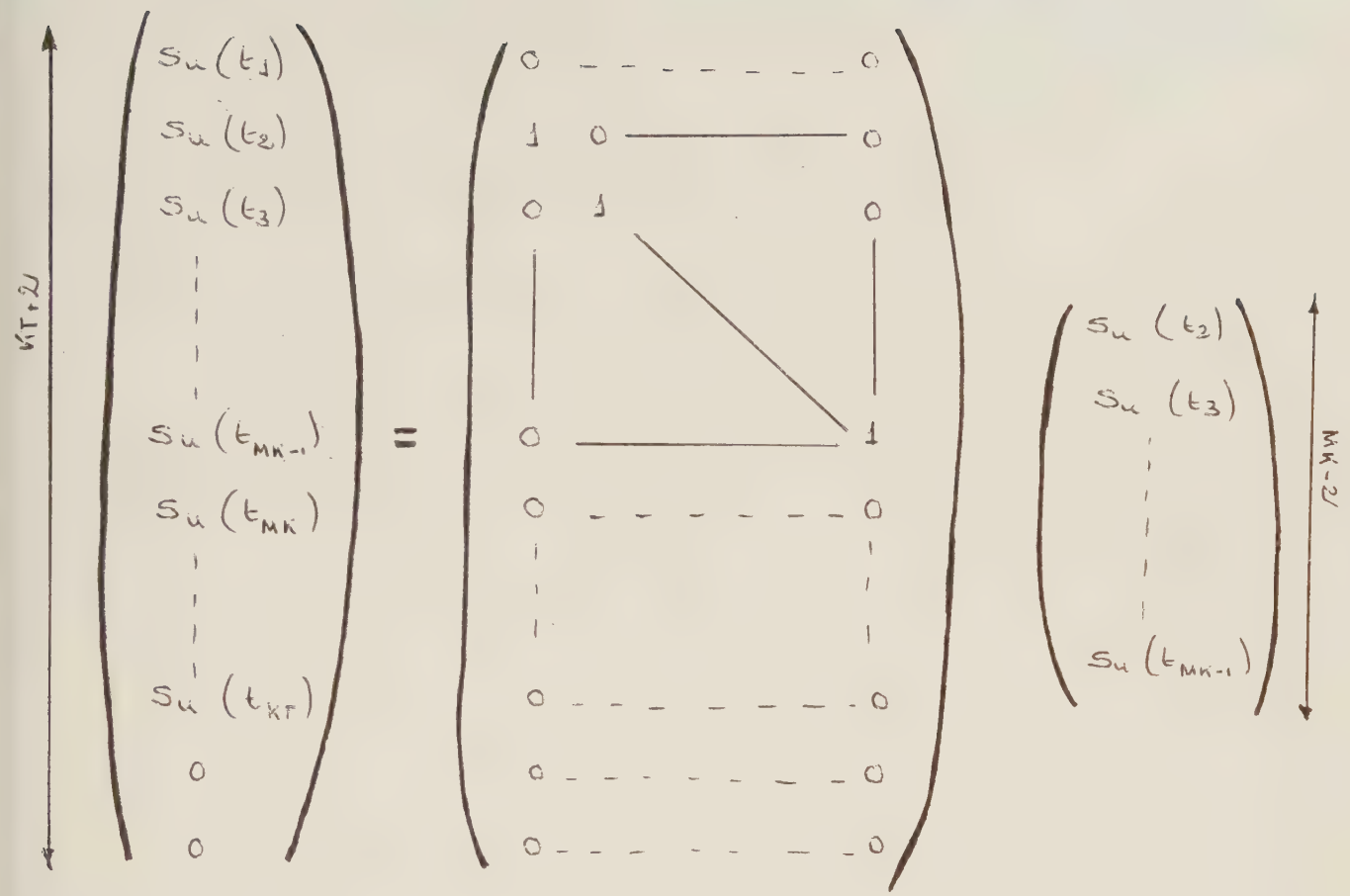


i.e. :

$$\boxed{T\,b = s} \qquad (*)$$

Les deux dernières lignes de la matrice T correspondent aux contraintes sur les paramètres exprimés par les équations (4) et (5).

2°- Mais, dans le cas qui nous intéresse ici, la fonction-spline naturelle S_u est nulle sur la seconde moitié de son support et aussi en son premier point de jonction (équations (3') et (2')). Ces contraintes se résolvent en pratiquant une immersion de $\mathbb{R}^{MK-2}$ dans $\mathbb{R}^{KT+2}$.



i.e. :

$$S = I_m S^*$$

où :

. I_m représente la matrice d'immersion de $^{MK-2}$ dans $^{KT+2}$.

. S* représente les valeurs non nulles de la spline en ses points de jonction.

3°- Par conséquent, les contraintes linéaires (2'), (3'), (4) et (5) sur les paramètres $(b_i)_{i=1,KT+2}$ s'écrivent :

$$b = T^{-1} I_m S^*$$

Finalement, si l'on pose :

$$\Phi(b) = \sum_{i=1}^{N} \left[\tilde{f}(x_i) - S_f(x_i, b) \right]^2$$

La recherche de la solution du problème :

$$\min_{b \in \mathbb{R}^{KT+2}} \Phi(b)$$

$$\cdot \sum_{l=1}^{KT+2} b_{2+l} = 0$$

$$\cdot \sum_{l=1}^{KT+2} t_l \, b_{2+l} = 0$$

$$\cdot \; S_u(t_l) \; 0$$

$$\cdot \; S_u(t_l) = 0 \qquad l = MK, KT$$

est équivalente à la recherche de la solution du problème :

$$\min_{S^* \in \mathbb{R}^{MK-2}} \Psi(S^*)$$

où

$$\Psi(S^*) = \sum_{i=1}^{N} \left[\tilde{f}(x_i) - S_f(x_i, (T^{-1} I_m)(S^*)) \right]^2$$

Cette minimisation dans $\mathbb{R}^{MK-2}$ (sans contrainte) sera effectuée par un sous-programme de la CISI, codé DMAGT1, et basé sur l'algorithme de D.W. MARQUARDT / 25 / (cf. Annexe II).

VI.2 - LES TESTS SUR DES SIMULATIONS MATHÉMATIQUES

VI.2.a - Les tests de débruitage

Sur un vecteur de données $(f(t_i))_{i=1,K}$ connues, on simule un bruit gaussien de variance $f(t_i)$ et d'espérance nulle en chaque valeur $f(t_i)$; on obtient :

$$y_i = f(t_i) + \varepsilon_i$$

$$\text{où } \varepsilon_i \sim \mathcal{N}(0, f(t_i))$$

$$\text{pour } i = 1,K$$

Et au vecteur de ces valeurs bruitées $(y_i)_{i=1,K}$ on applique la méthode de débruitage par fonctions-splite polynômiales.

VI.2.b - Les tests de déconvolution instrumentale

On construit un vecteur $u = (u_i)_{i=1,N_{INF}}$ d'informations "test" et un vecteur $g = (g_i)_{i=1,N_{INST}}$ de "valeurs de la fonction-instrument".

Puis on calcule leur convolée séquentielle en appliquant une méthode d'intégration par trapèzes :

$$f_1 = 0$$

$$f_{i+1} = h \sum_{p=1}^{i} g_{i-p+2} \left(\frac{u_p}{3} + \frac{u_{p+1}}{6}\right) + g_{i-p+1} \left(\frac{u_p}{6} + \frac{u_{p+1}}{3}\right)$$

$$i = 1, N_{EXP-1}$$

et on applique à (f_i) ainsi définie, la méthode de déconvolution instrumentale par fonctions-spline.

On fait aussi des tests en simulant un bruit sur la convolée f (comme il est dit au paragraphe précédent) et en appliquant successivement à cette fonction bruitée, débruitage puis déconvolution.

VI.2.c - Les tests d'autodéconvolution

De la même façon que dans la déconvolution instrumentale, on construit un vecteur $u = (u_i)_{i=1,N_{INF}}$ d'informations "test".

Puis on calcule l'autoconvolée, de cette fonction test, soit par trapèzes, soit analytiquement si le calcul n'est pas trop lourd, et on applique à la fonction f ainsi obtenue la méthode d'autoconvolution par fonctions-spline.

On fait aussi des essais d'autodéconvolution sur la même fonction après l'avoir bruitée puis débruitée.

VI.2.d. - Nature des fonctions tests

Tous les tests que nous avons faits, ont été effectués sur des fonctions de nature trigonométrique.

VII - CONCLUSION

Le travail que nous venons de présenter ici avait pour but la mise au point, sur des simulations mathématiques, de méthodes de déconvolution instrumentale et d'autodéconvolution lorsque l'on se trouve en présence de bruit gaussien non stationnaire.

Ces méthodes se divisent en deux parties :

1°- Le débruitage

2°- La déconvolution instrumentale
ou
L'autodéconvolution

Au niveau du débruitage, nous avons mis en place une méthode déjà efficiente dans le cas d'un bruit stationnaire, qui nécessite une approximation (sur la trace de A_λ (cf. § II.2.b)) et dont la légitimité d'emploi n'a pas été démontré. Cependant, dans le cadre de l'échantillonnage des fonctions (même fortement bruitées) que nous avons étudiées, les différentes perturbations que nous avons pu faire subir à la trace de A_λ n'ont eu aucune influence tangible sur la qualité finale du lissage.

Nous obtenons des résultats tout à fait satisfaisants, comme l'illustrent les courbes G1 a/b et G2 a/b.

Pour ce qui concerne la déconvolution instrumentale, de nombreuses autres méthodes existent déjà, mais celle que nous

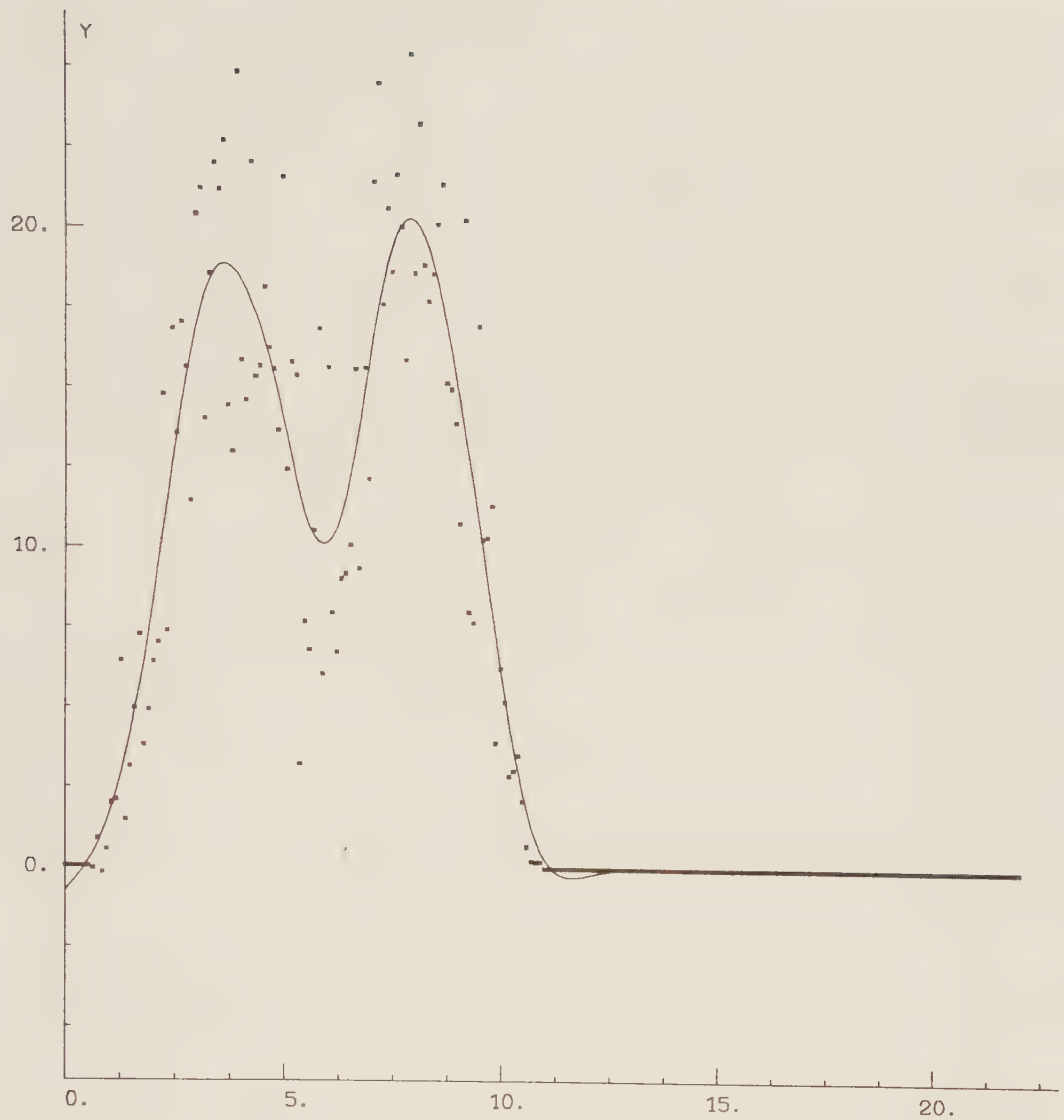
proposons ici offre des résultats fort bons, même dans le cas de données entâchées d'un bruit important (cf. les courbes G3 a/b et G4 a/b qui représentent la déconvolution instrumentale de mêmes spectres dont le rapport signal/bruit est différent).

La résolution d'équations d'autoconvolution, quoique pouvant apparaître comme un cas particulier de la résolution d'équations de convolution, s'est avérée (du fait de la non-linéarité du problème) d'une sensibilité nettement plus marquée aux variations sur les données. Mais, grâce à la qualité du lissage réalisé par la méthode de débruitage que nous avons proposée, les résultats d'autoconvolution que nous avons obtenus sont tout à fait satisfaisants tant que le rapport signal/bruit garde une signification appréciable (cf. les courbes G8 a/b, G9 a/b, G10 a/b, G11 a/b et G12 a/b/c).

La méthode d'autodéconvolution dont la validité a ainsi été prouvée sur des modèles simulés, peut être appliquée à des spectres expérimentaux autoconvolés dont on possède toutes les valeurs. Malheureusement, il arrive très souvent que les expériences faites ne permettent pas de récupérer le spectre entier. Il convient alors de faire une estimation arbitraire de la longueur du support total de l'autoconvolée afin de pouvoir appliquer notre méthode d'autodéconvolution.

TESTS DE DEBRUITAGE

Débruitage

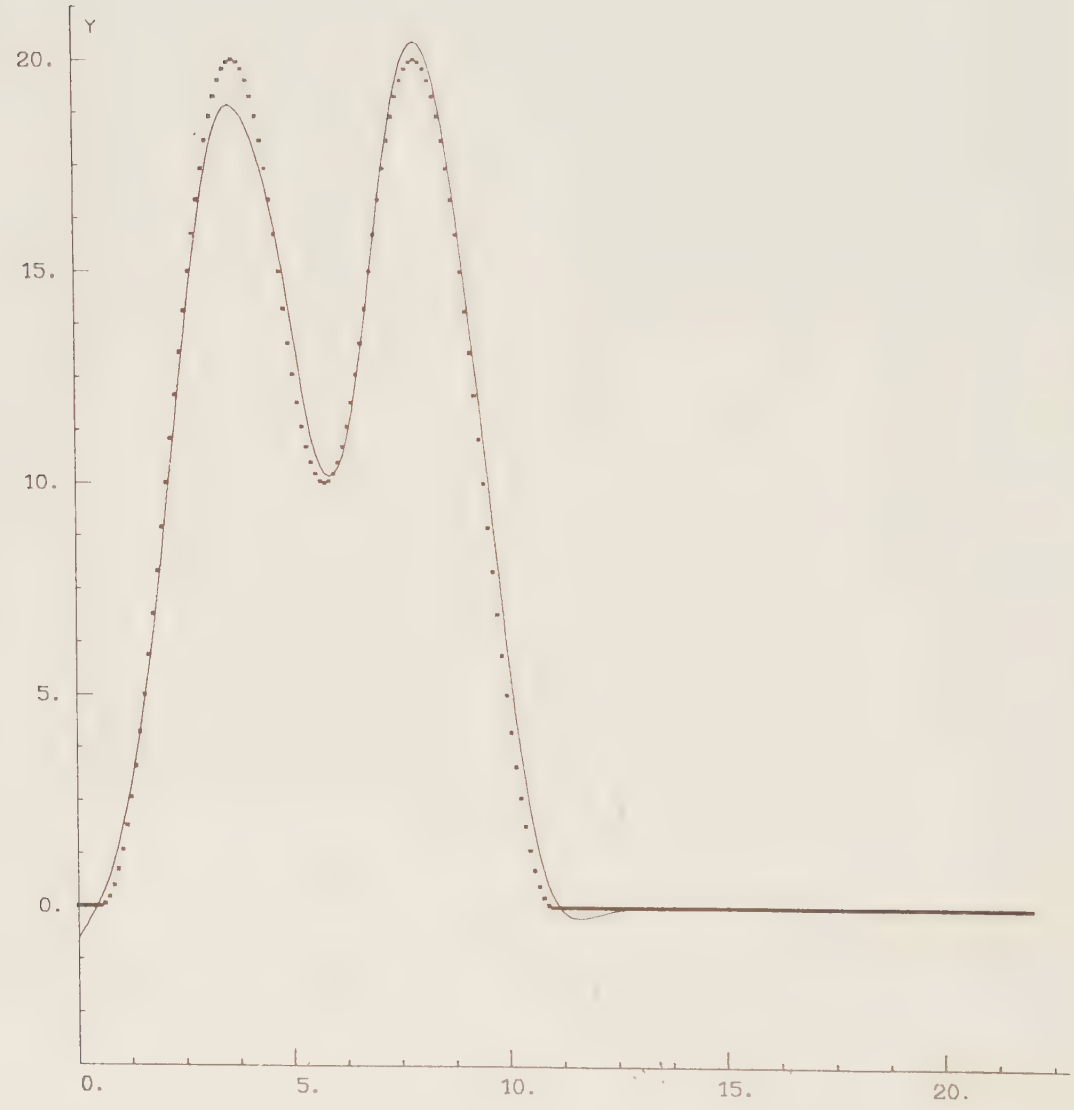


.... Courbe bruitée f = u + bruit

—— Courbe débruitée

G.1 a

Comparaison entre la fonction test avant bruitage et la courbe débruitée.

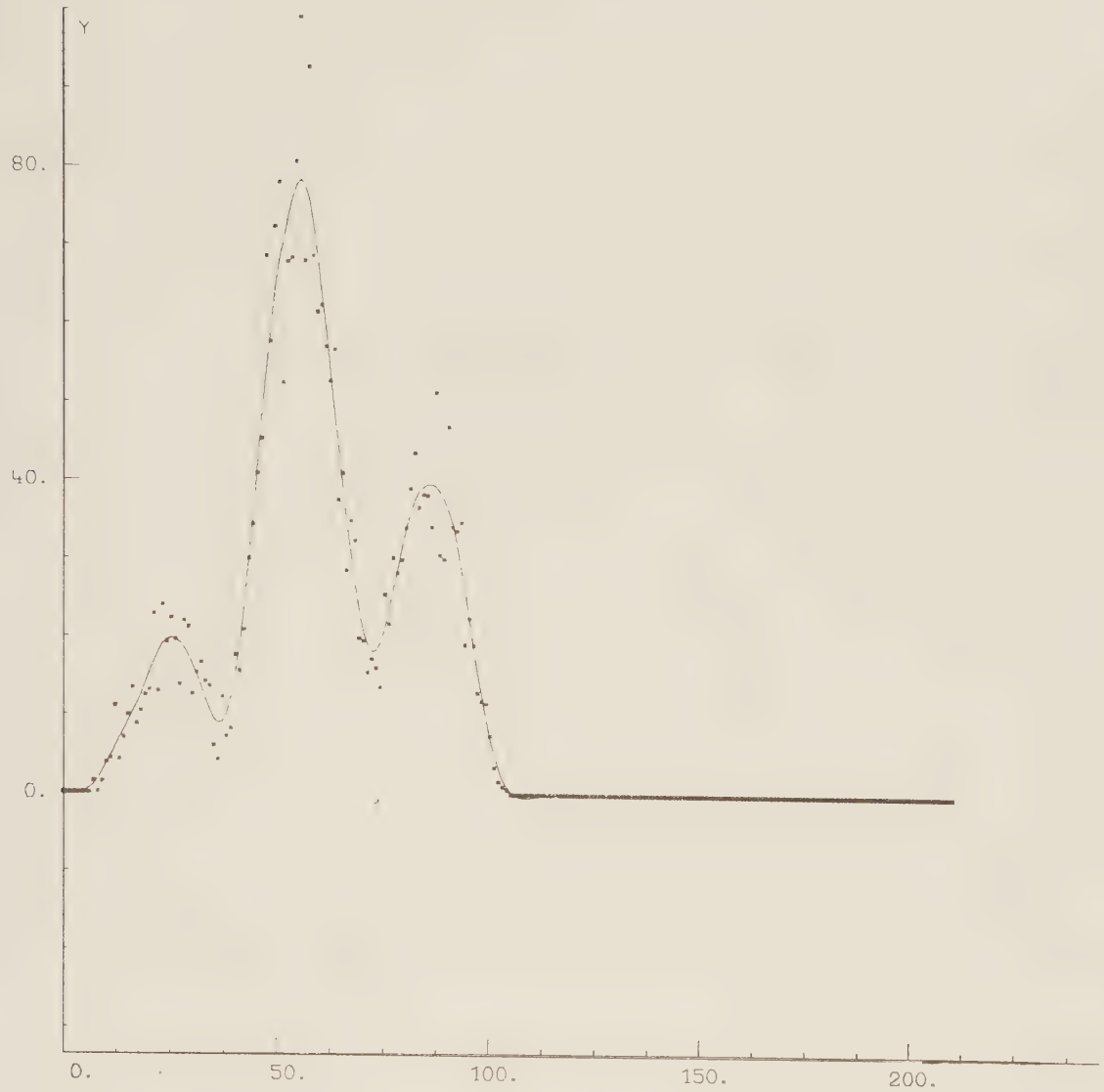


•••• Courbe test u

——— Courbe obtenue par débruitage.

G.1 b

Débruitage

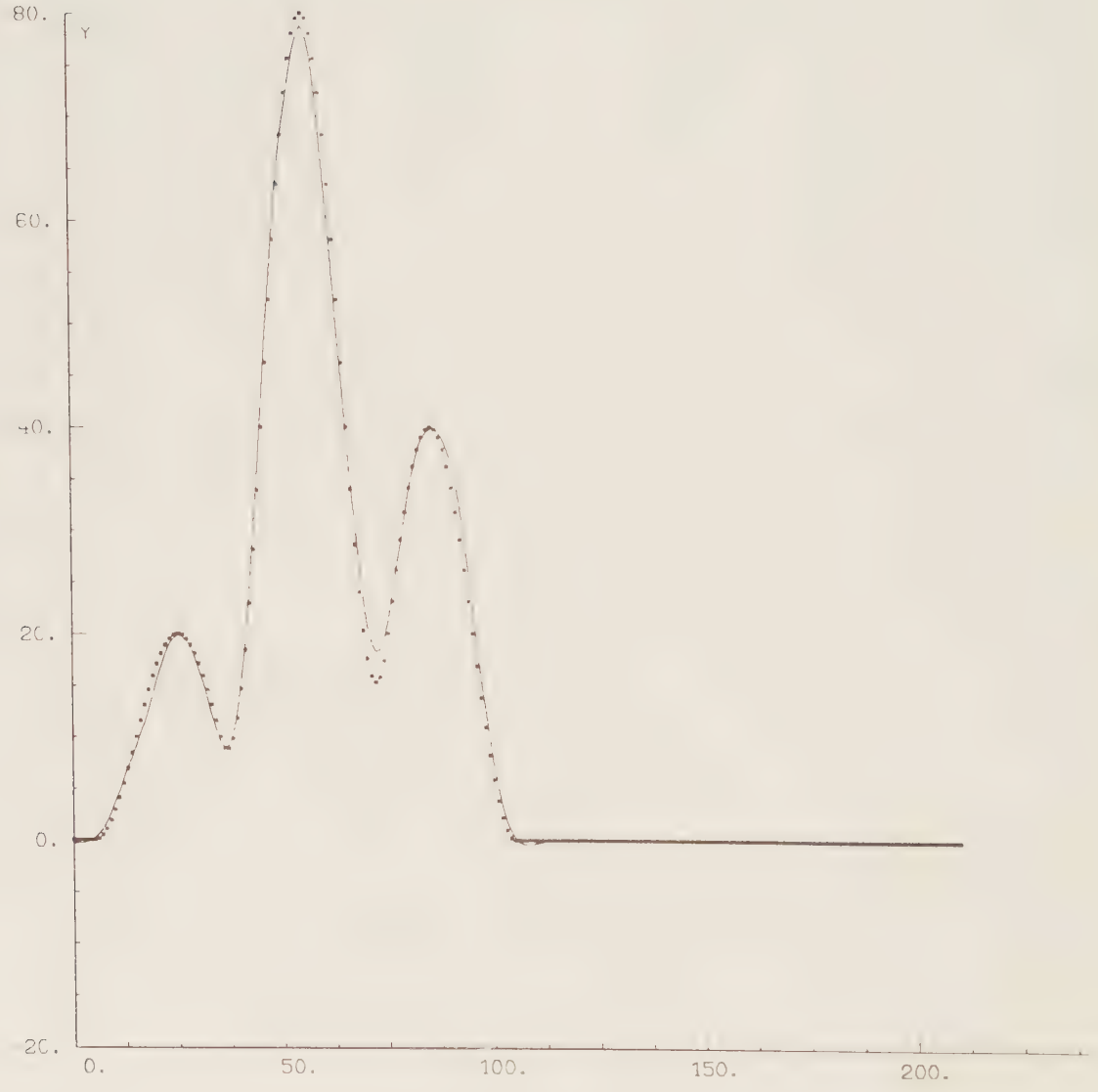


· · · · Courbe bruitée f = u + bruit

——— Courbe débruitée

G. 2 a

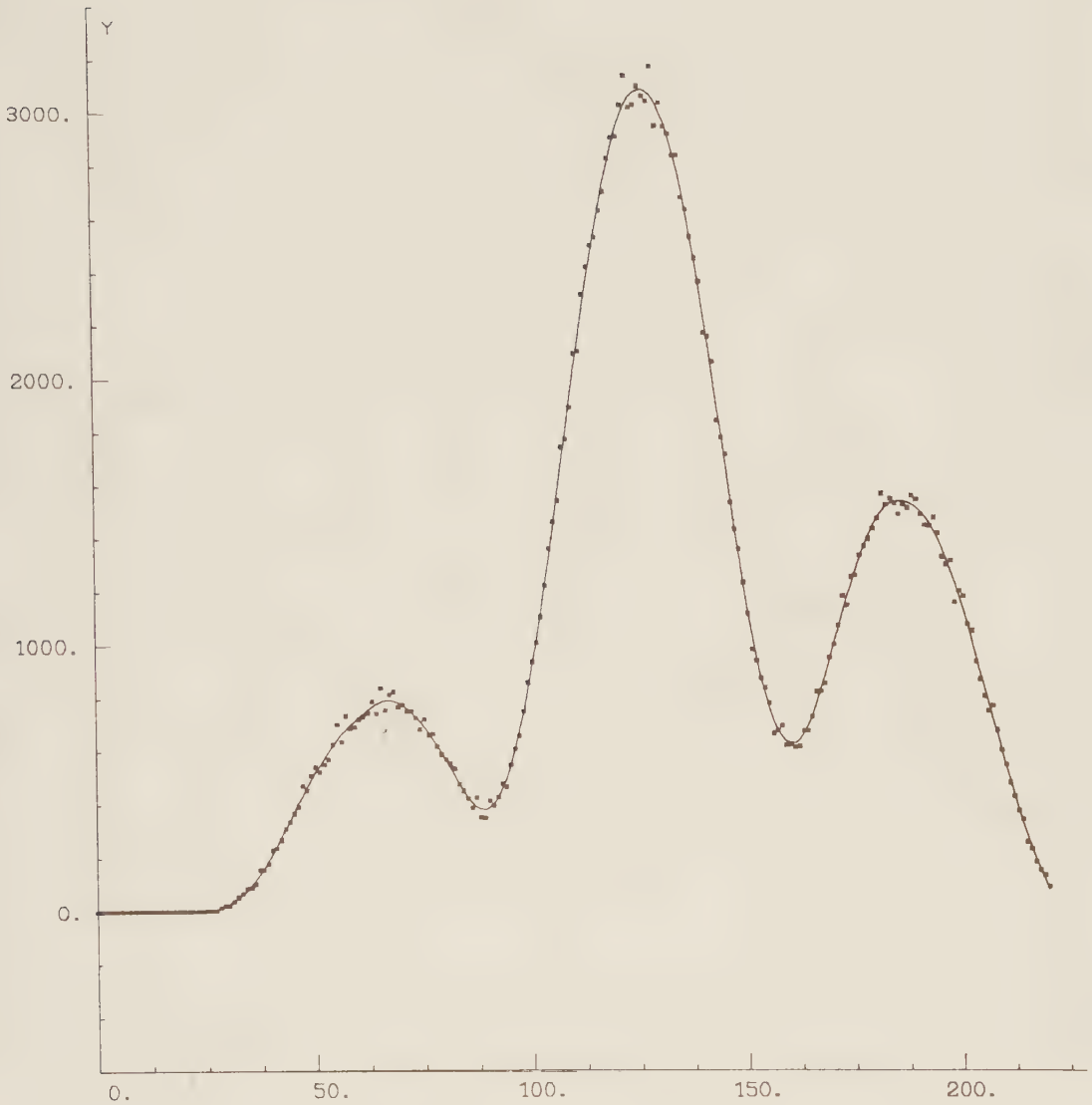
Comparaison entre la fonction test avant bruitage et la courbe débruitée.



G.2 b

TESTS DE DECONVOLUTION INSTRUMENTALE AVEC BRUIT

Débruitage d'une convolée instrumentale.

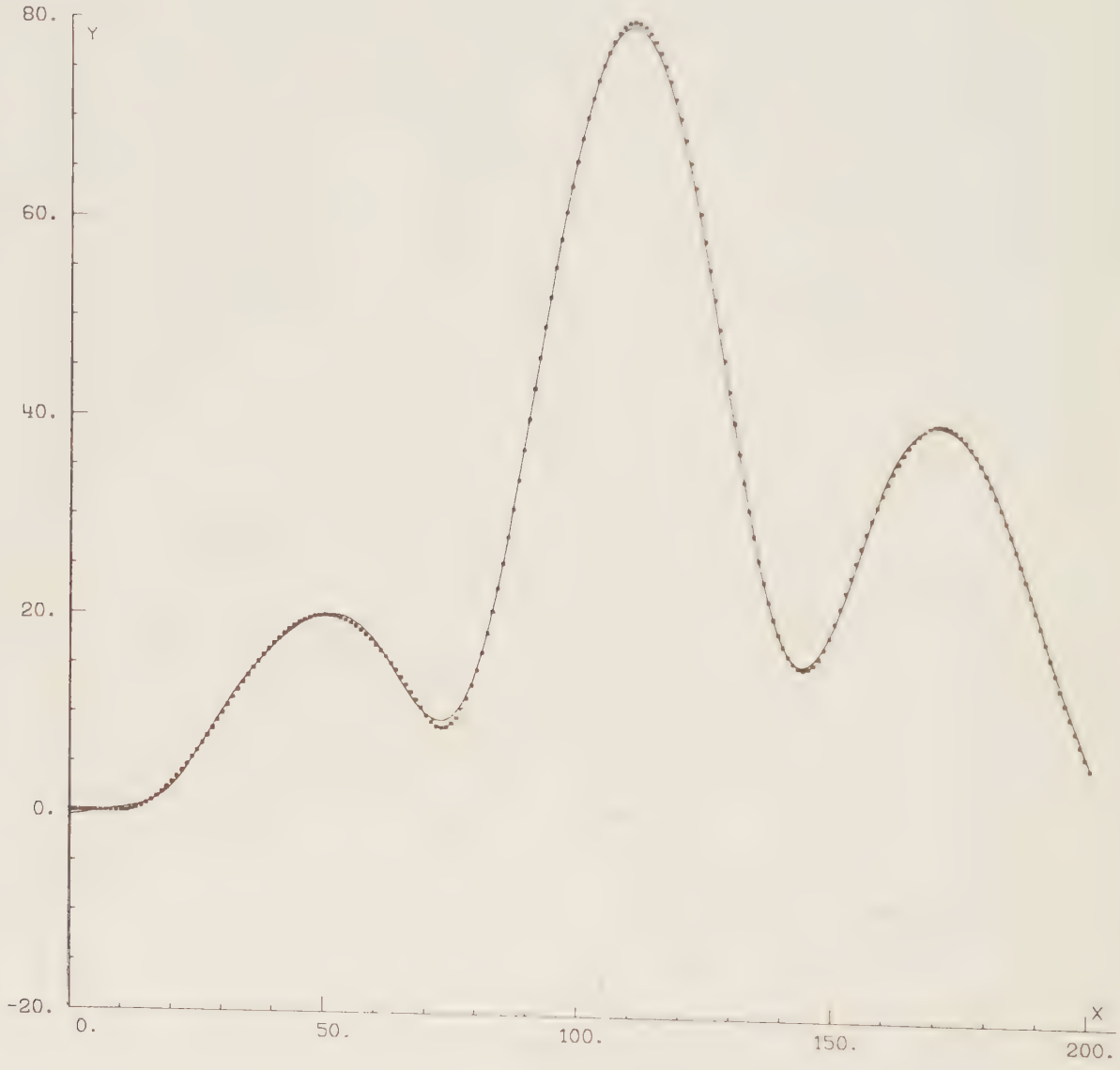


. . . . Valeurs expérimentales : fbr = u * g + bruit

——— Courbe obtenue par débruitage

G.3.a

Déconvolution instrumentale d'une fonction bruitée.

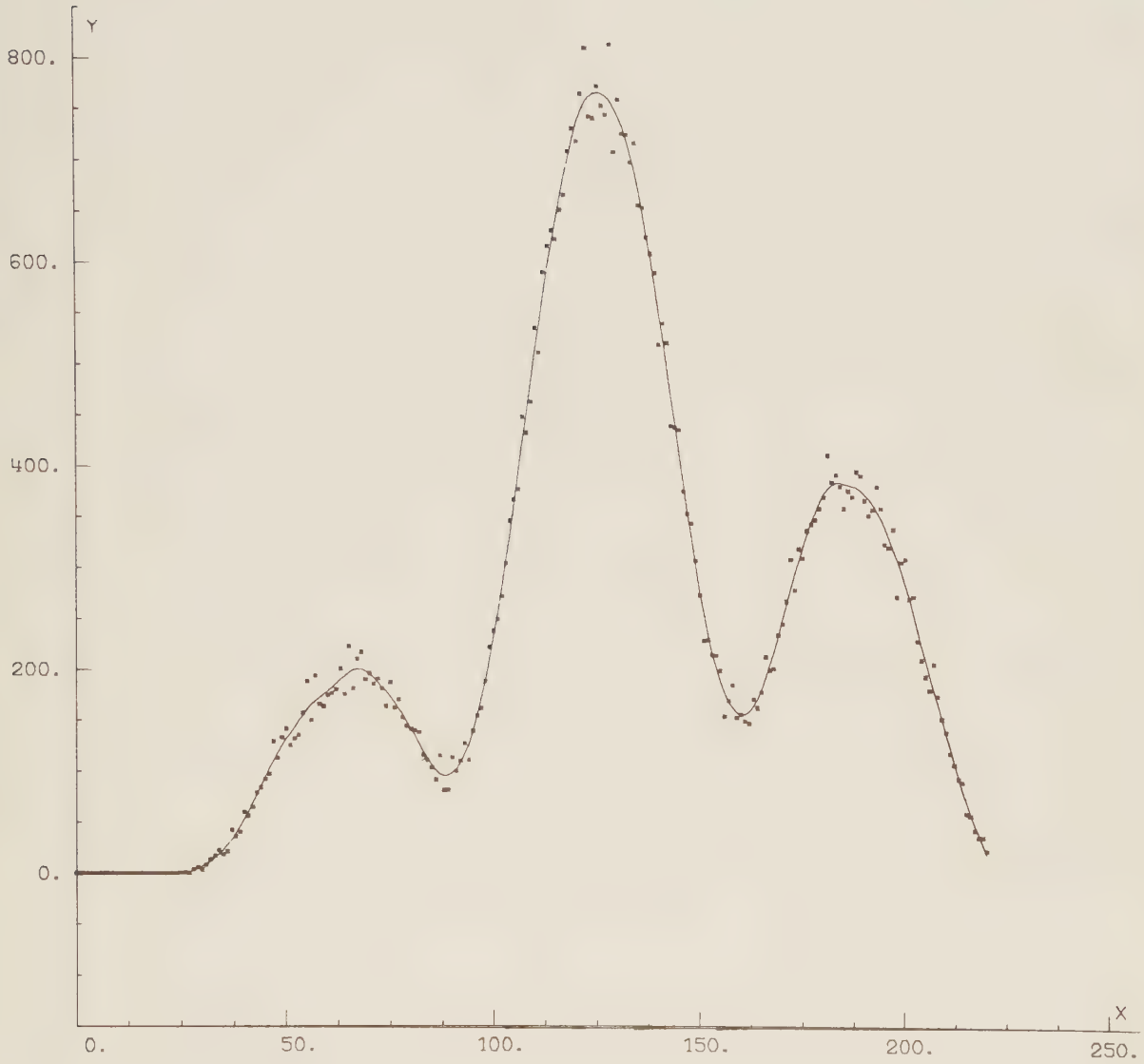


• • • • Courbe test u.

——— Courbe obtenue par déconvolution.

G.3.b

Débruitage d'une convolée instrumentale.

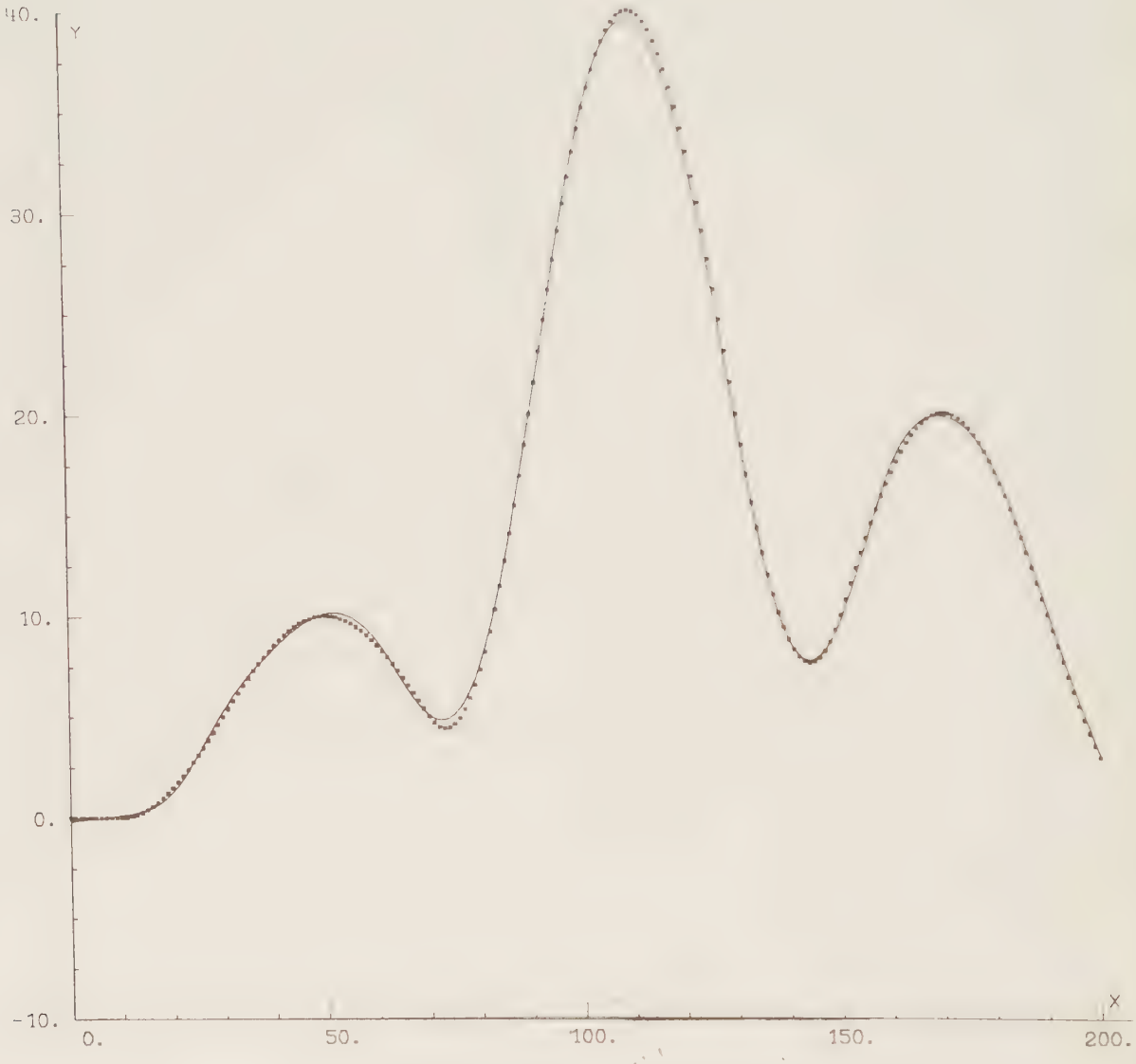


. . . . Convolée instrumentale bruitée.

——— Lissage.

G.4 .a

Déconvolution instrumentale d'une fonction bruitée.



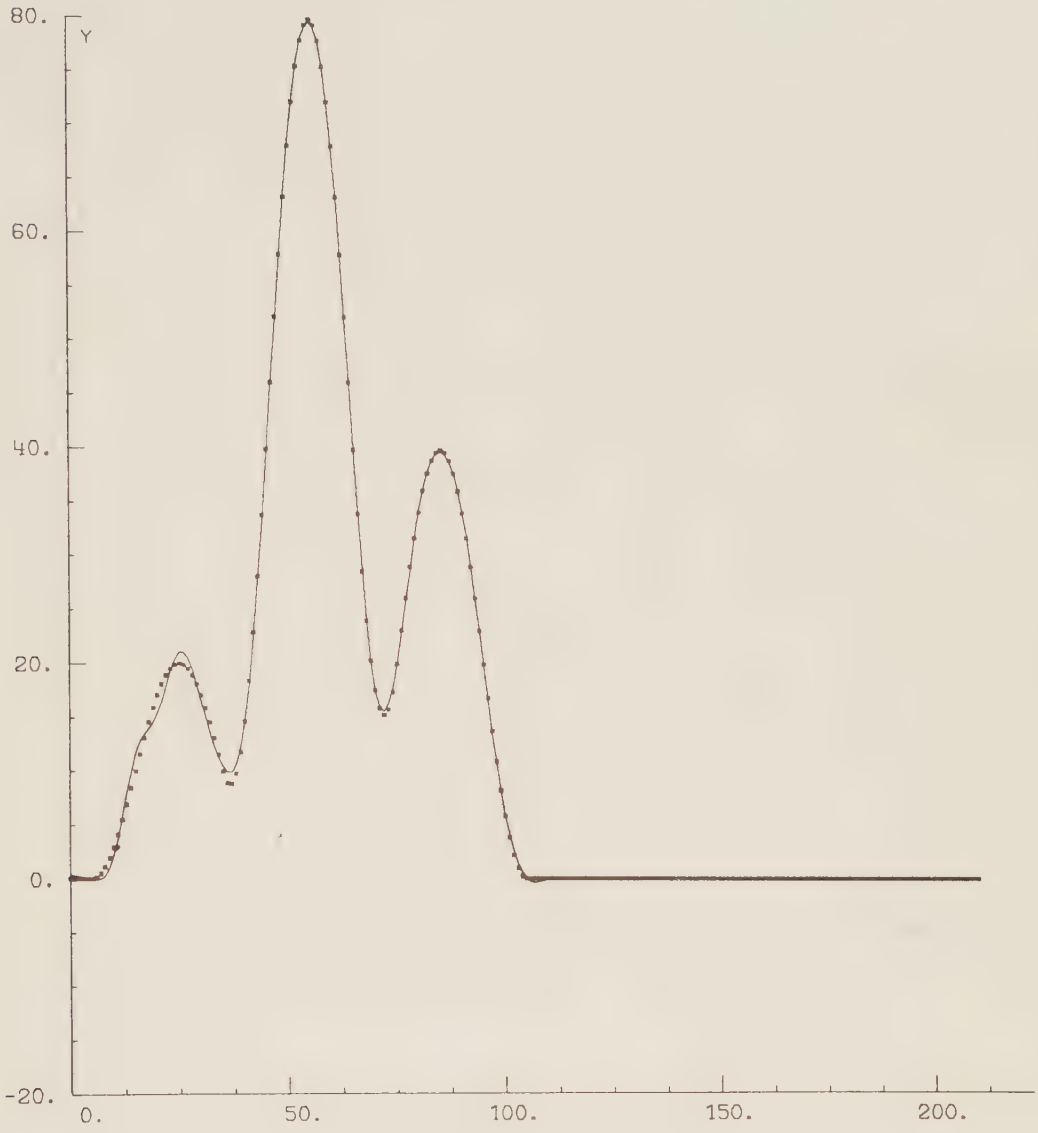
• • • • Courbe test u.

——— Courbe obtenue par déconvolution instrumentale.

G. 4.b

TESTS D'AUTODECONVOLUTION SANS BRUIT

Autodéconvolution d'une fonction non bruitée.

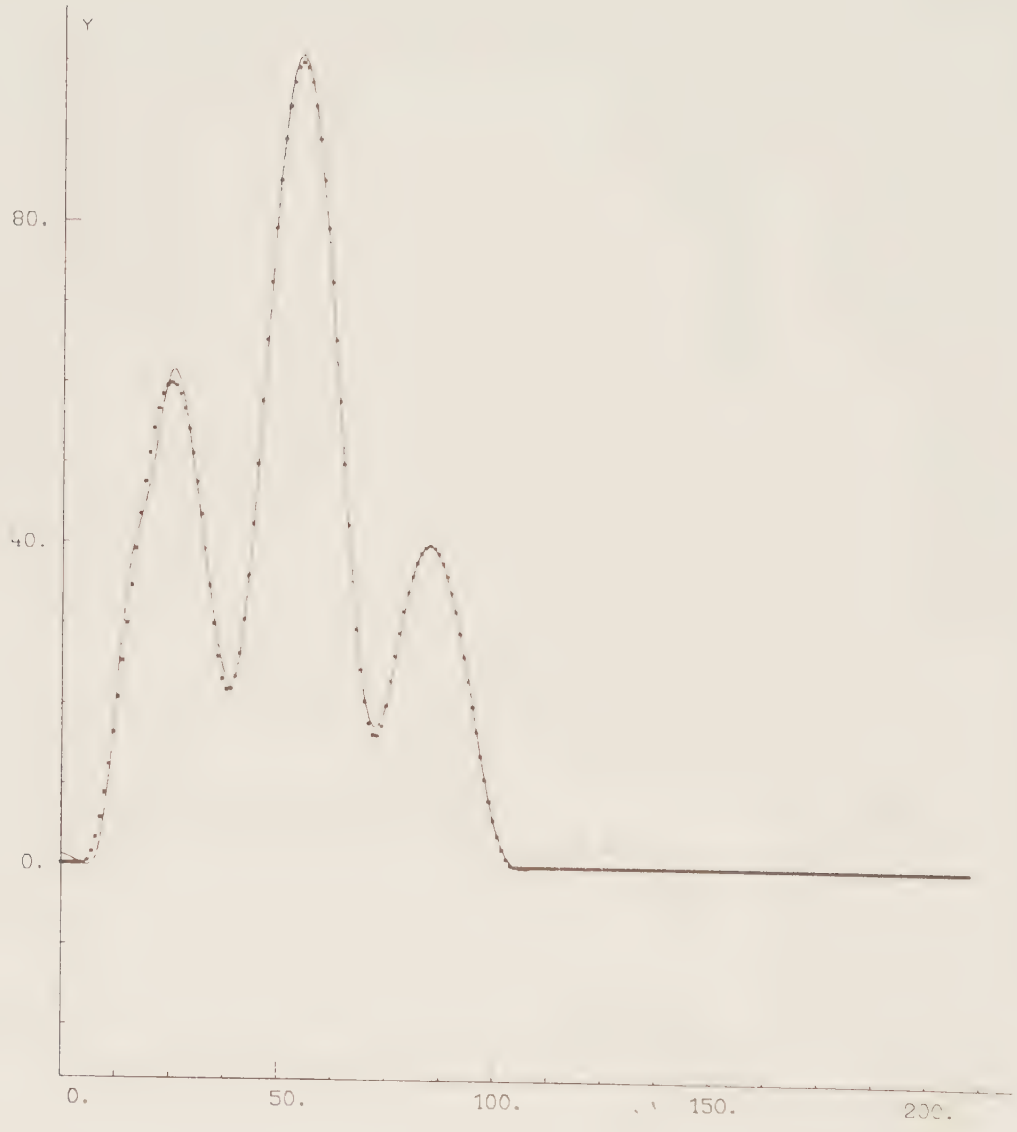


• • • • Courbe test u.

——— Courbe obtenue par autodéconvolution.

G.5

Autodéconvolution d'une fonction non bruitée.

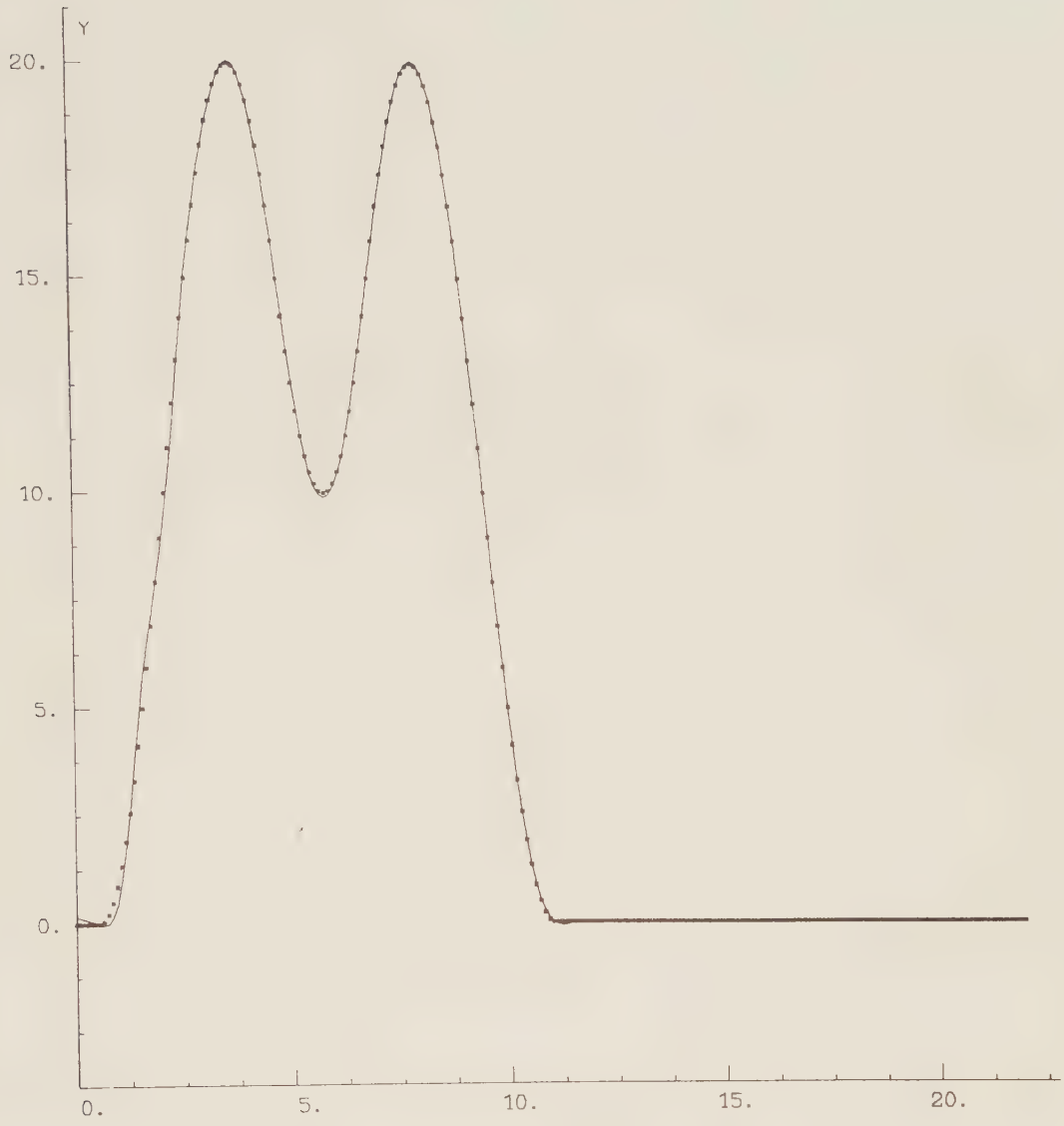


.... Courbe test u.

——— Courbe obtenue par autodéconvolution.

G. 6

Autodéconvolution d'une fonction non bruitée.



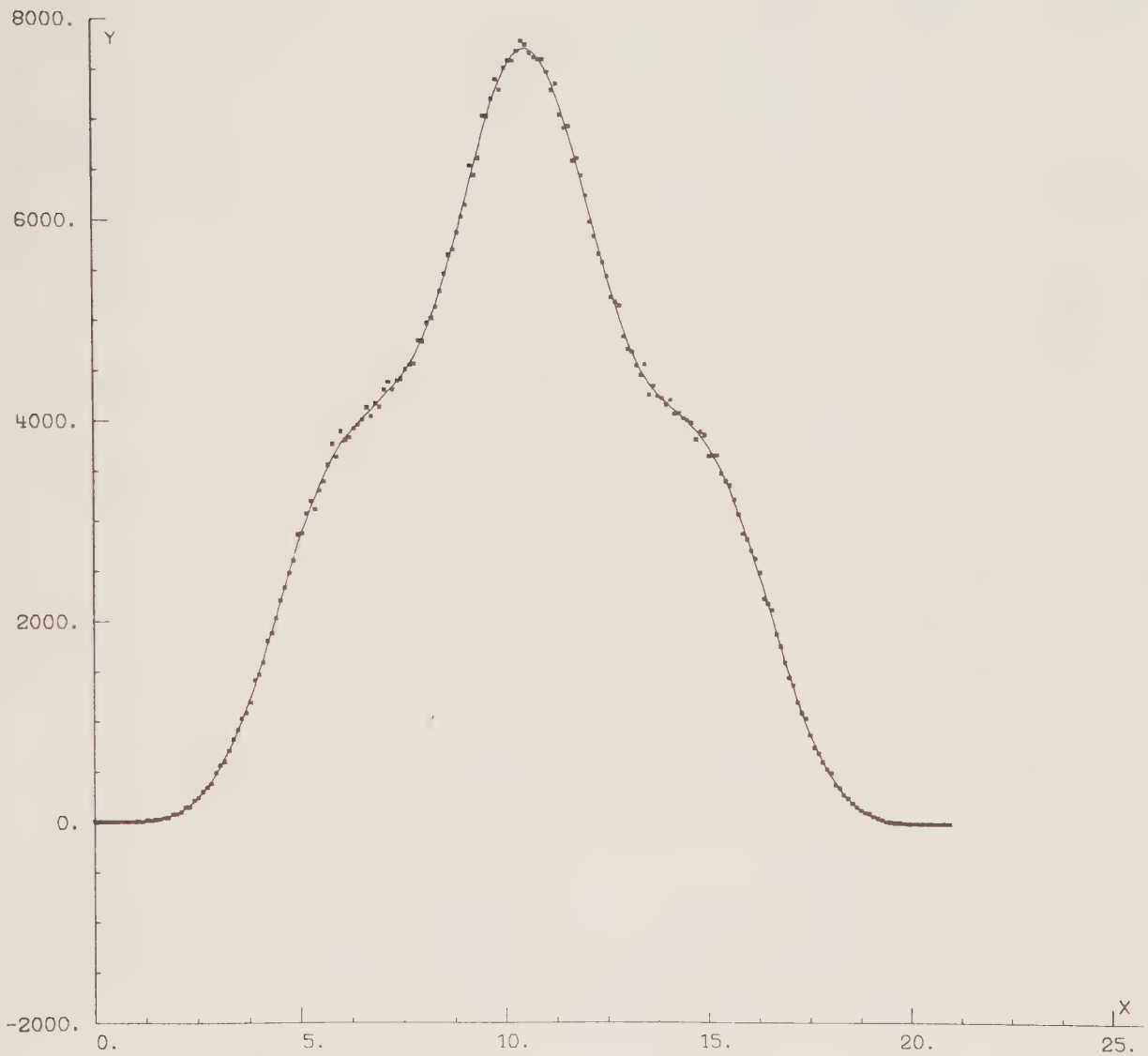
. . . . Courbe test u.

——— Courbe autodéconvolée.

G.7

TESTS D'AUTODECONVOLUTION AVEC BRUIT

Débruitage d'une autoconvolée.

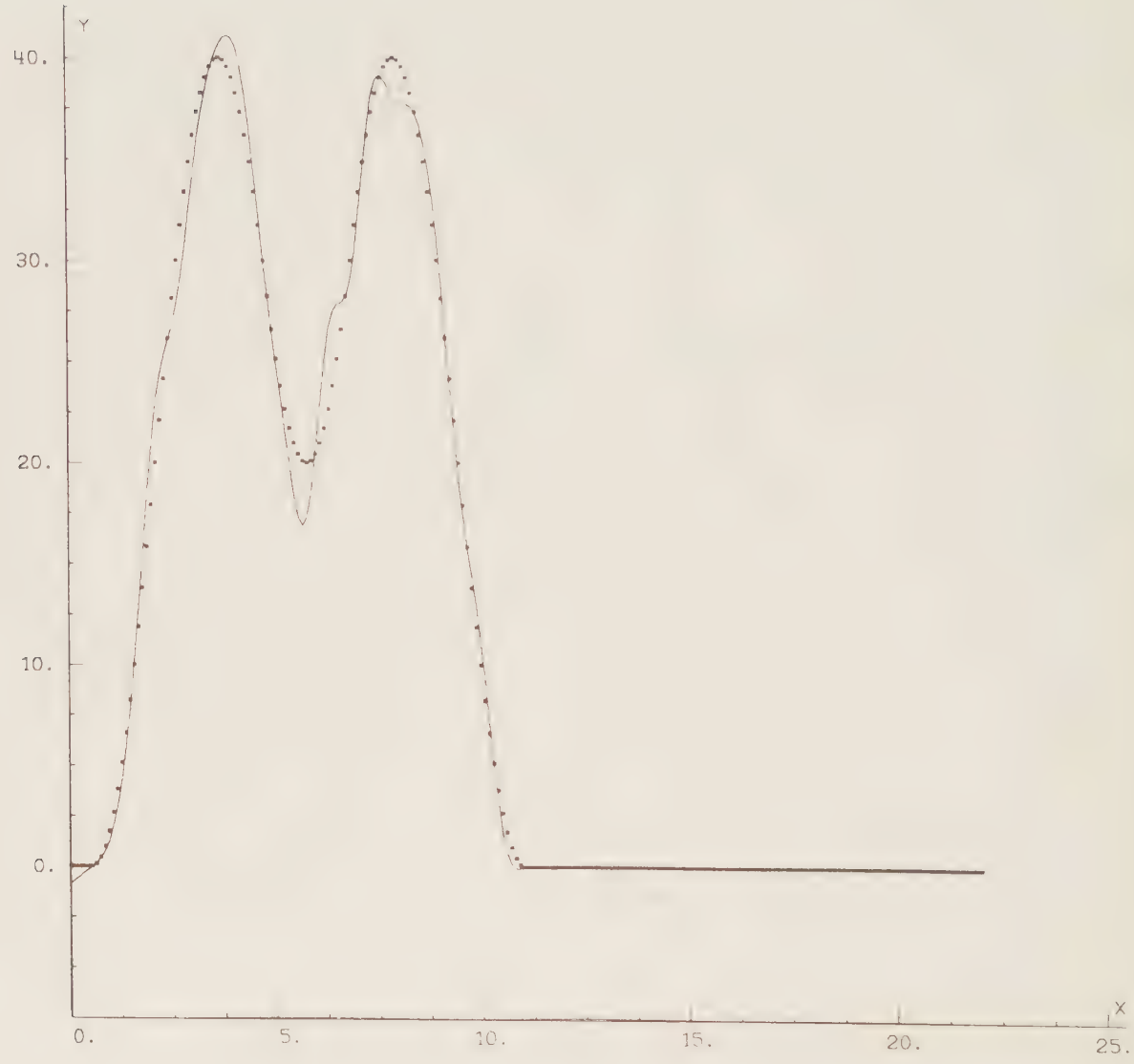


. . . . Courbe expérimentale.

——— Courbe obtenue par débruitage.

G. 8.a

Autodéconvolution d'une fonction bruitée.

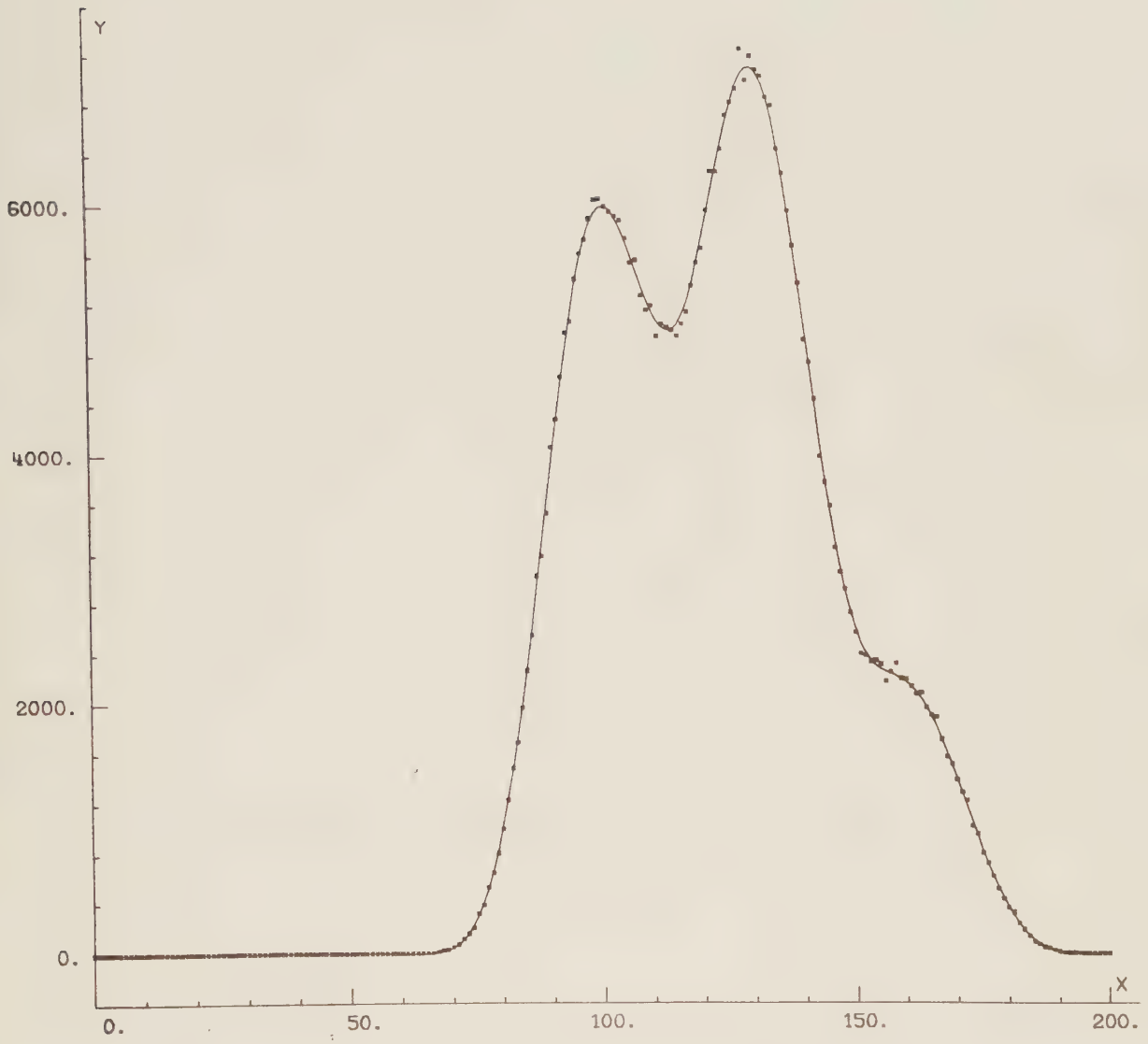


• • • • Courbe test u.

——— Courbe obtenue par autodéconvolution.

G.8.b

Débruitage d'une autoconvolée.

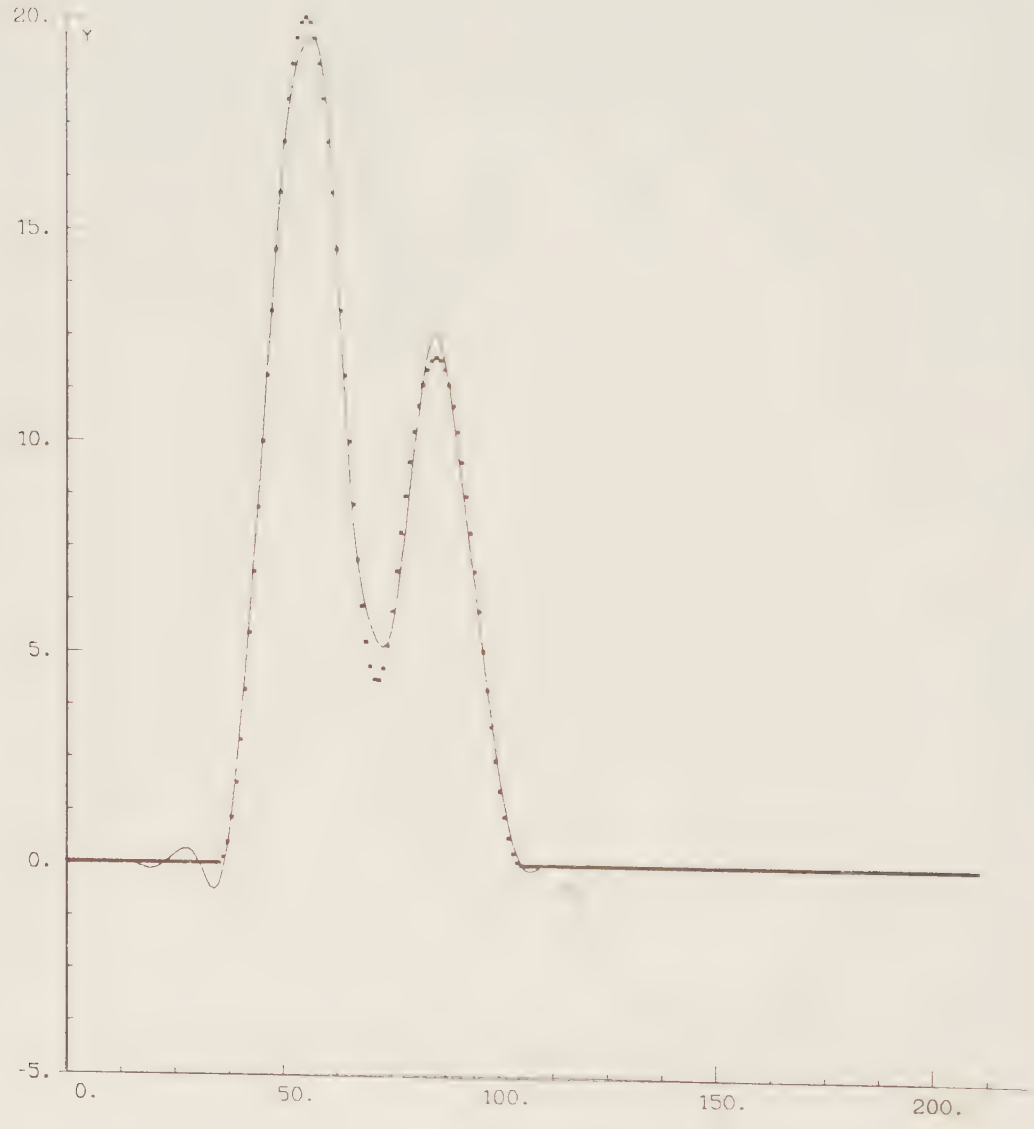


. . . . Courbe expérimentale.

——— Courbe obtenue par débruitage.

G. 9.a.

Autodéconvolution d'une fonction bruitée.

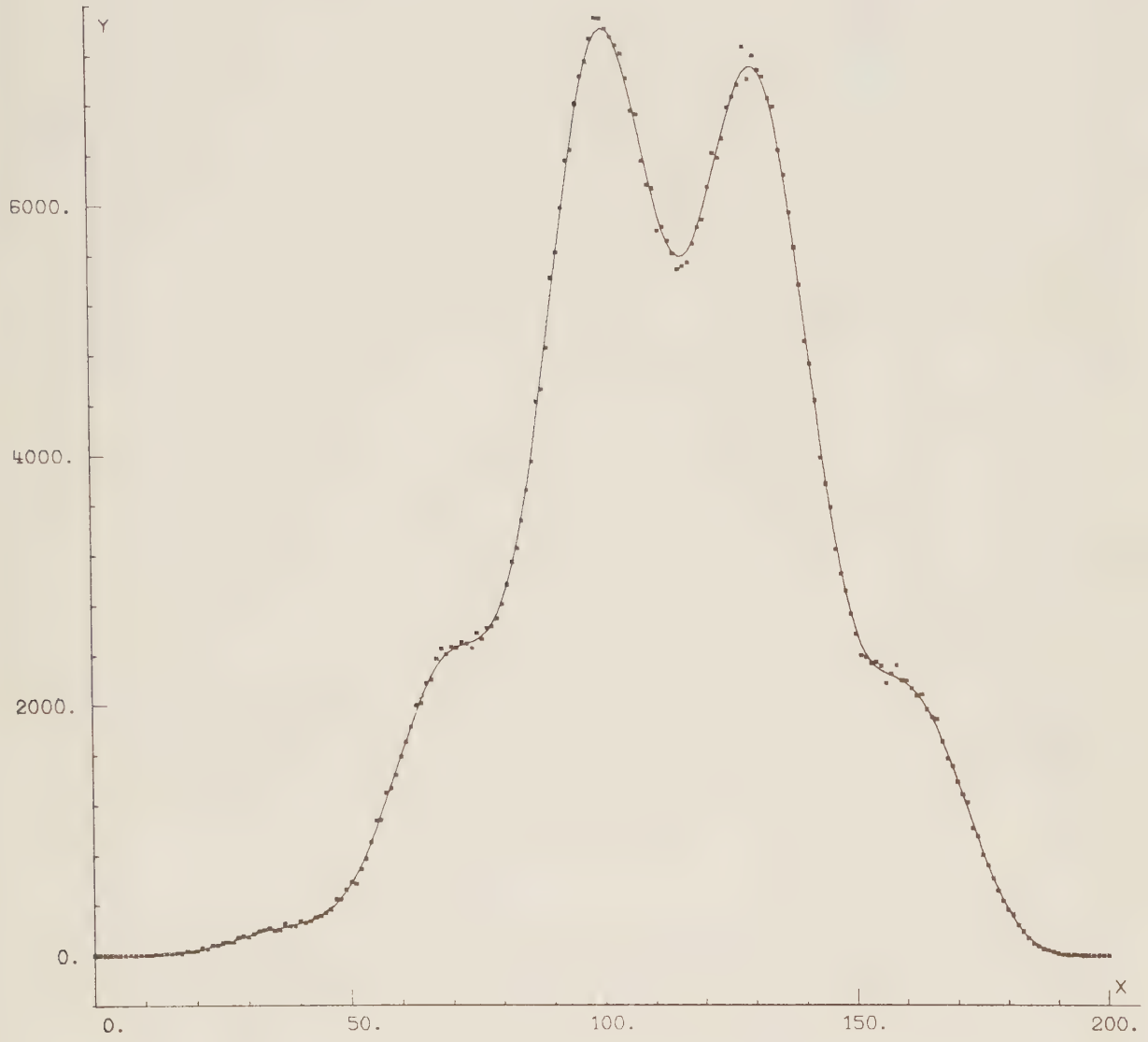


• • • • Courbe test u.

——— Courbe obtenue par autodéconvolution.

G.9.b.

Débruitage d'une autoconvolée.

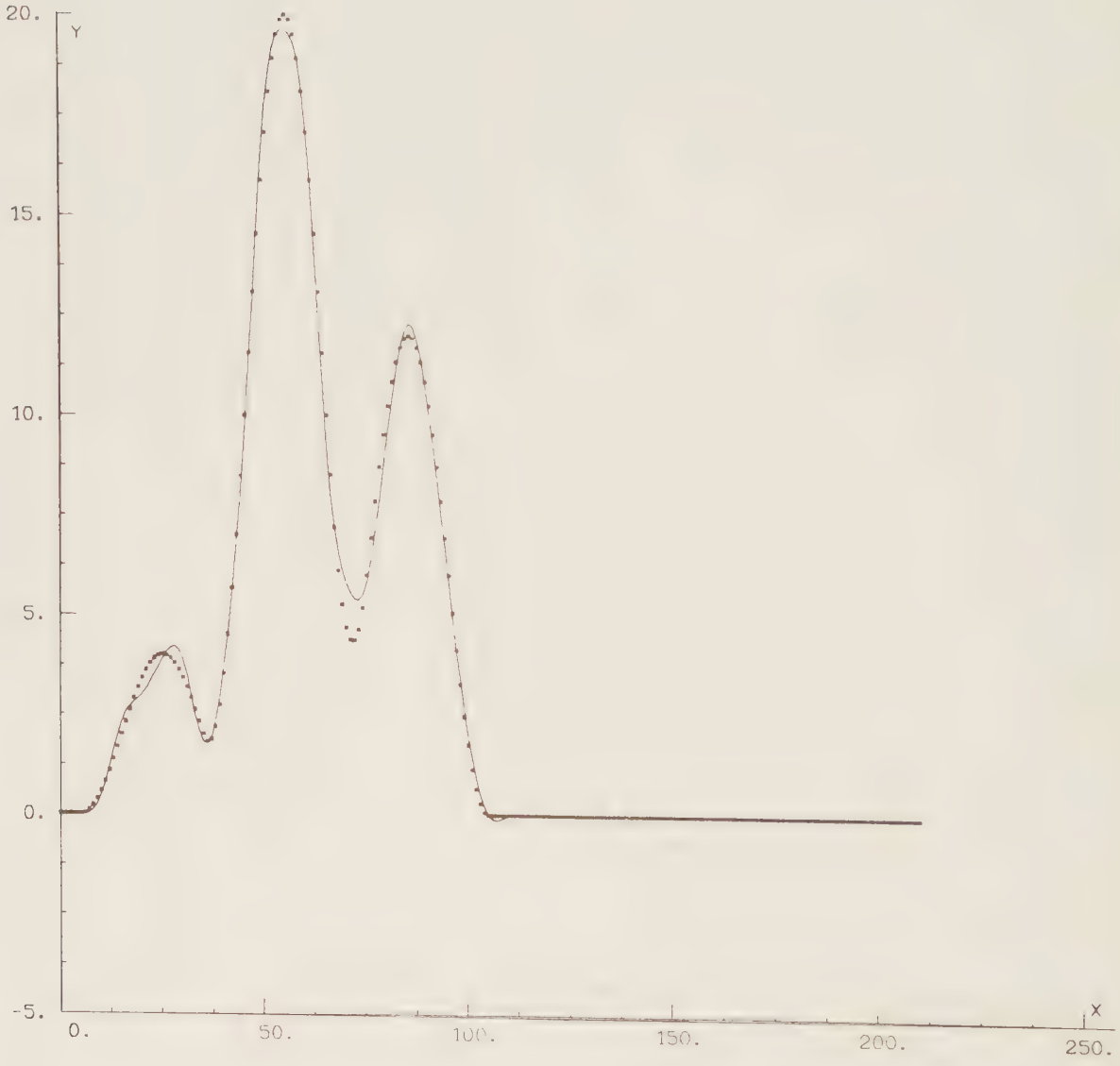


• • • • Courbe expérimentale.

——— Courbe obtenue par débruitage.

G.10.a.

Autodéconvolution d'une fonction bruitée.

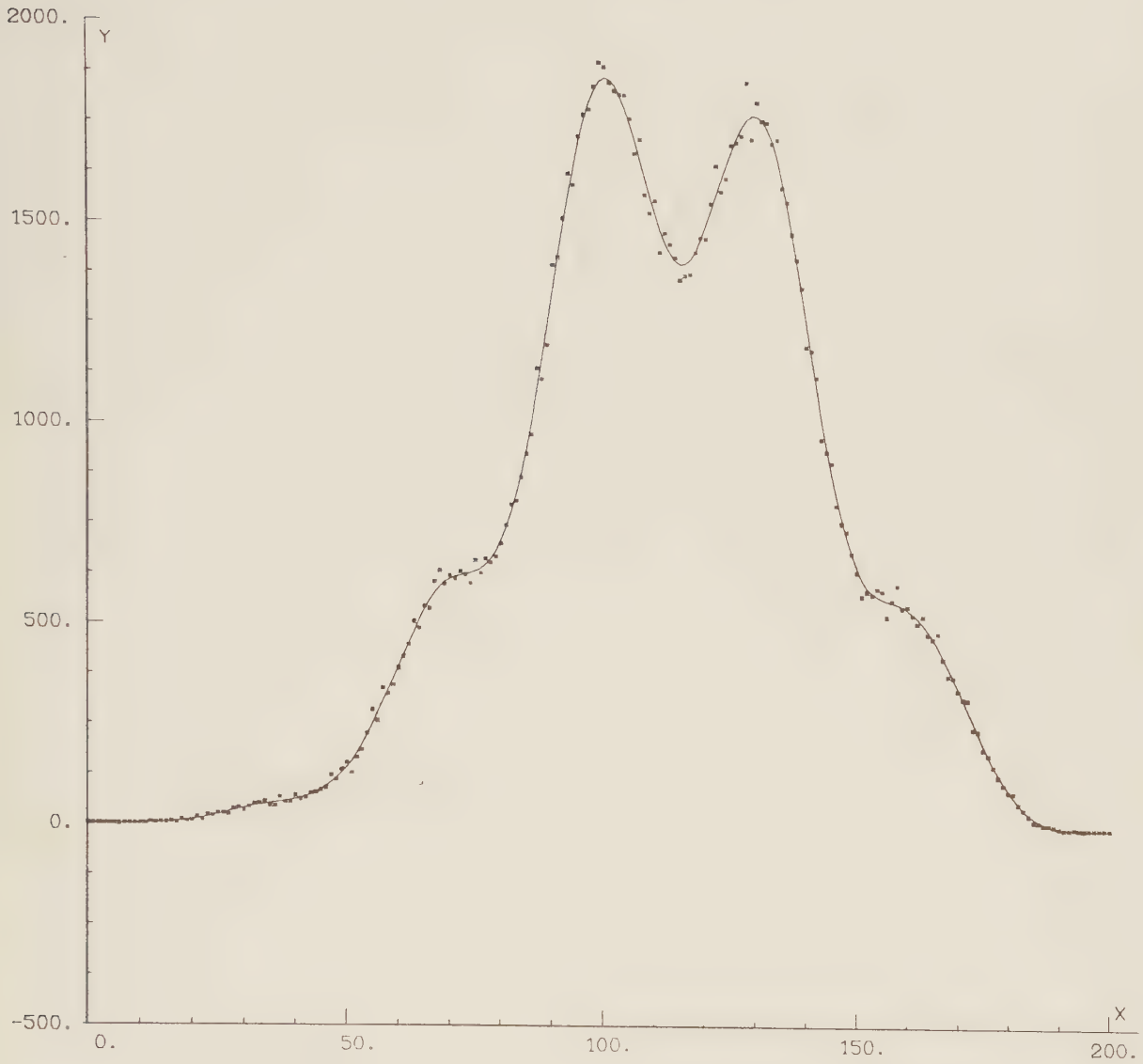


• • • • Courbe test u.

——— Courbe obtenue par autodéconvolution.

G.10.b

Débruitage d'une autoconvolée.

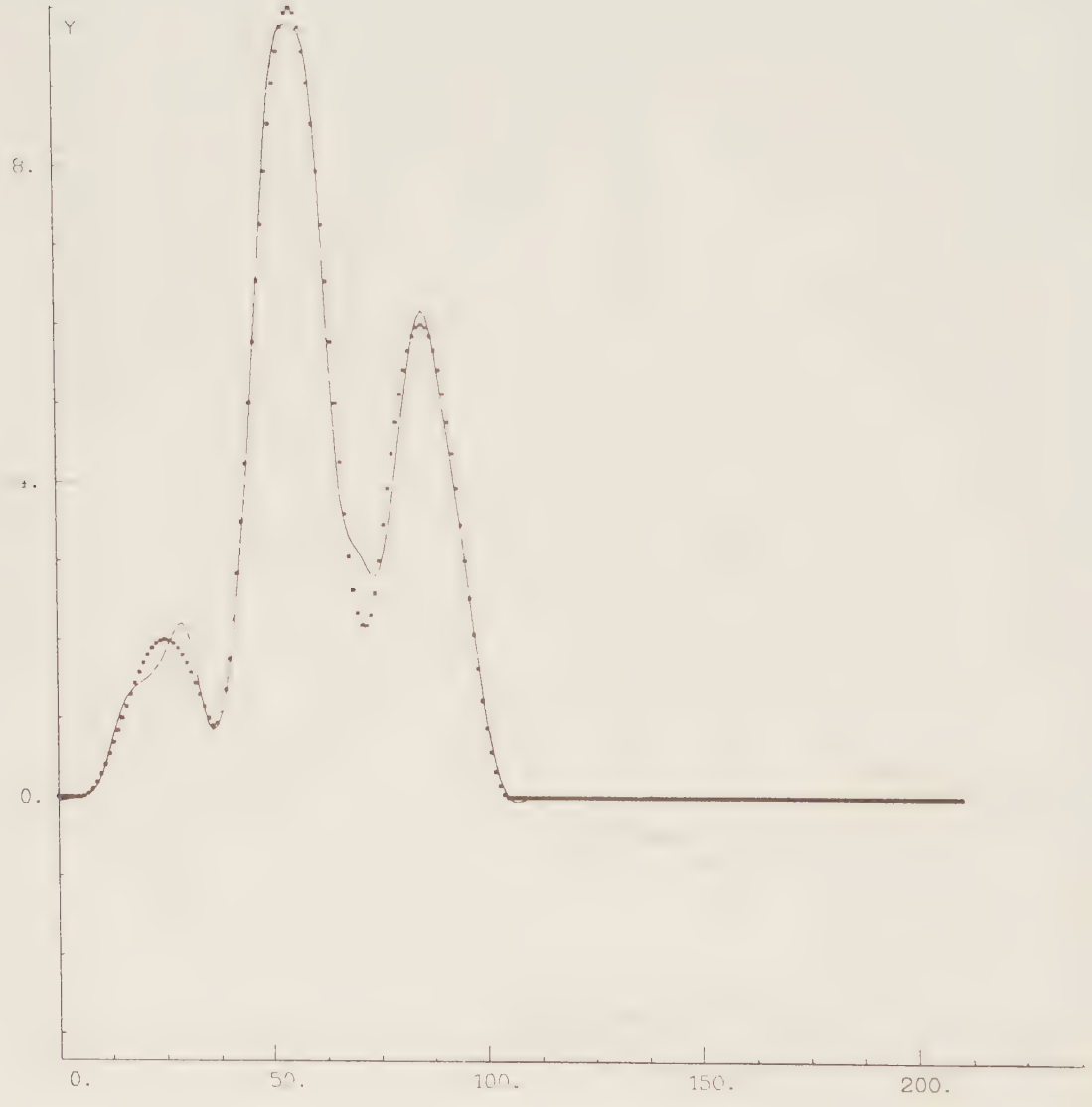


. . . . Courbe expérimentale.

——— Courbe obtenue par débruitage.

G.11.a

Autodéconvolution d'une fonction bruitée.

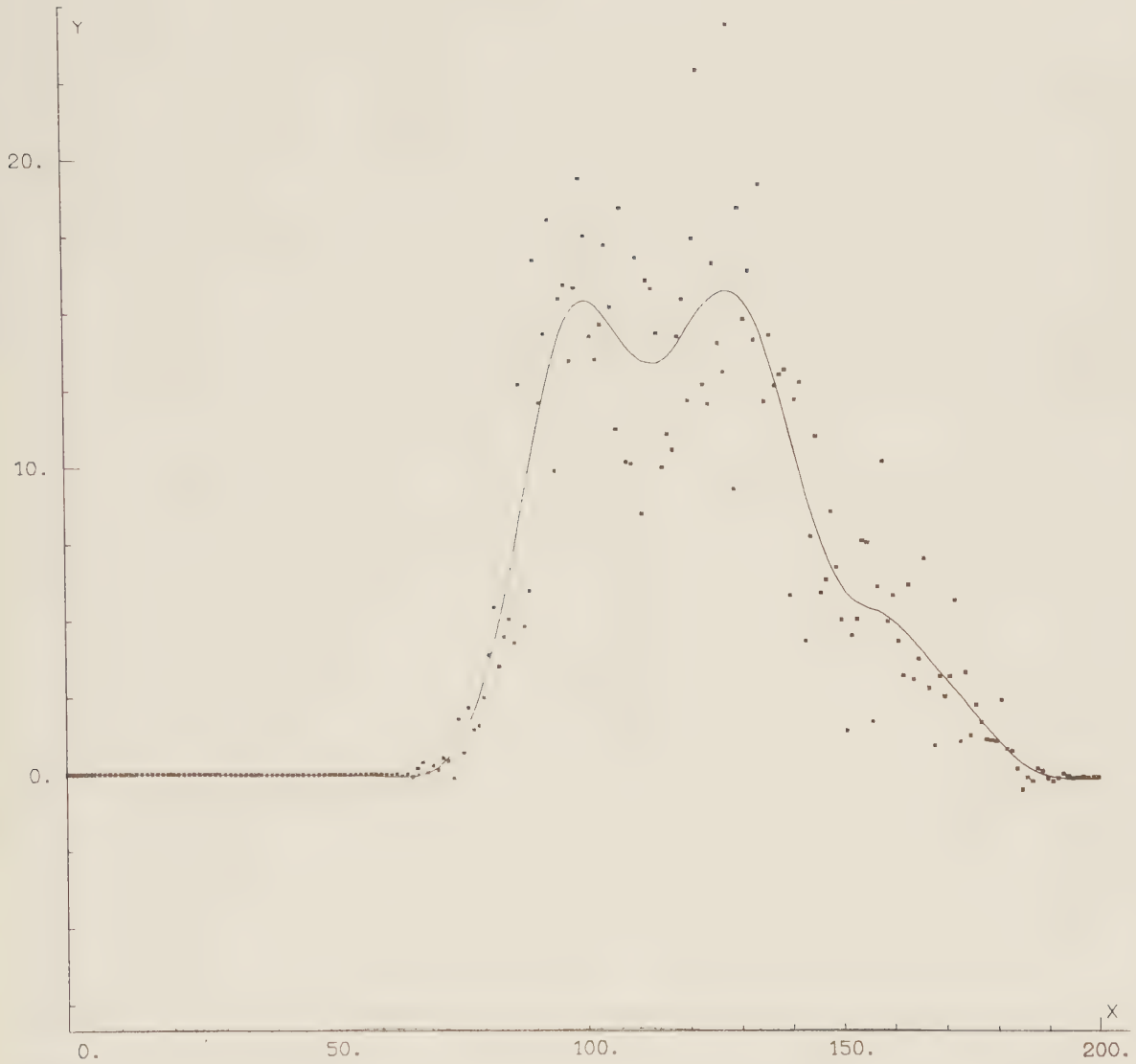


. . . . Courbe test u.

——— Courbe obtenue par autodéconvolution.

G.11.b

Débruitage d'une autoconvolée

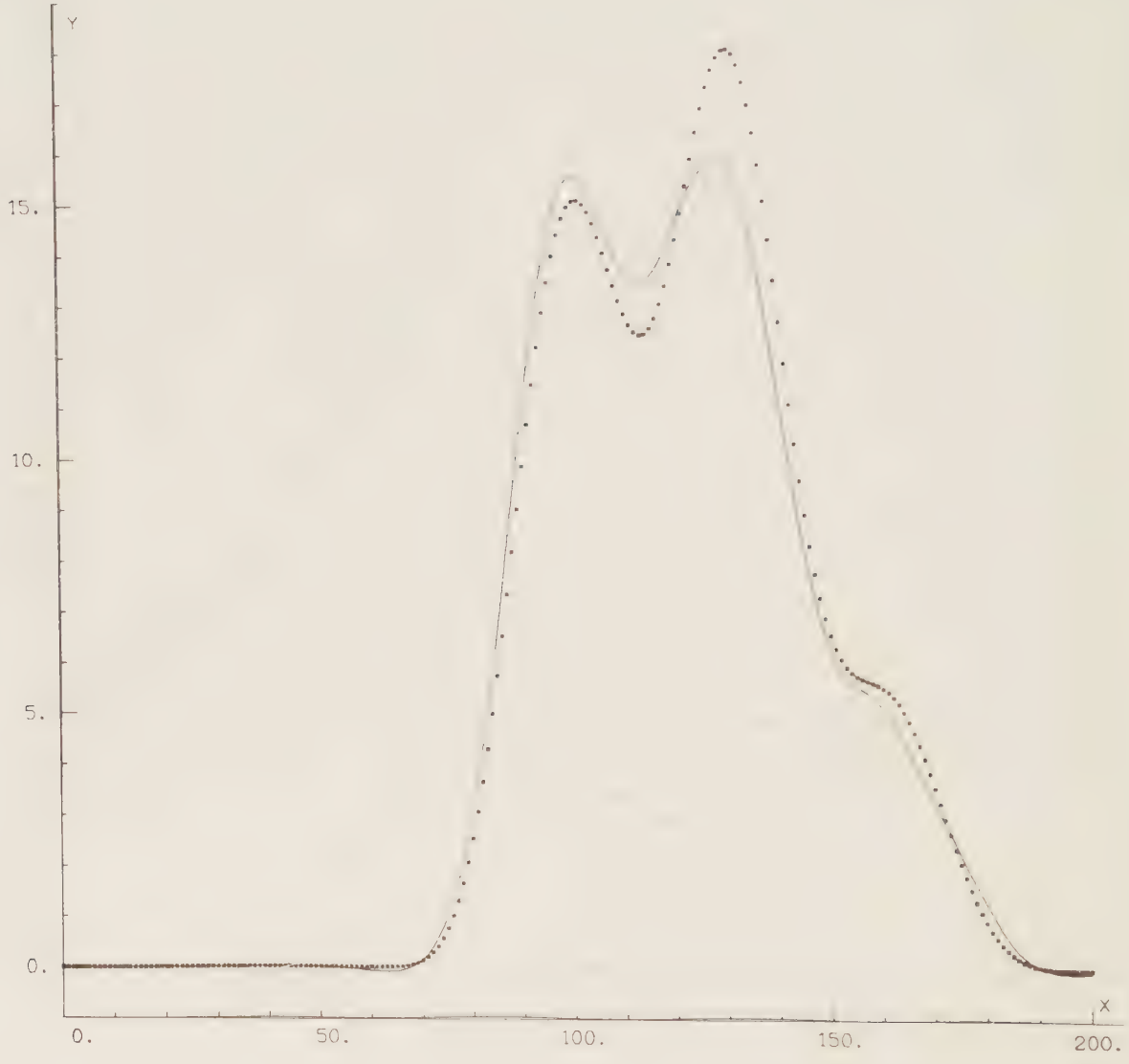


• • • • Autoconvolée bruitée.

——— Lissage.

G.12.a

Comparaison entre l'autoconvolée test et la fonction obtenue par débruitage.



. . . . Autoconvolée avant bruitage.

——— Lissage.

G.12.b

Autodéconvolution d'une fonction très fortement bruitée.

• • • • Fonction est u.

——— Courbe obtenue par autodéconvolution.

G.12.c

BIBLIOGRAPHIE

/ 1 / C. BOIZIAU
"Interaction Between Metastable Rare Gas Atoms and Surfaces".
"Inelastic Particle-Surface Collisions".
Springer Verlag, 1981.

/ 2 / C. DE BOOR
"A Practical Guide to Spline".
Springer Verlag, 1978.

/ 3 / J. CEA
"Optimisation".
Dunod, 1970.

/ 4 / P. CRAVEN and G. WAHBA
"Smoothing Noisy Data with Spline Functions".
Estimating the correct degrée of smoothing by the method of generalized cross-validation.
Num. Math. n° 31 (tome 4), 1979. (377 → 403).

/ 5 / C. GAROT
Thèse Université Paris-Sud. (Orsay 01.06.1978).

/ 6 / G. KIMELDORF and G. WAHBA
"A correspondance between Bayesian Estimation on Stochastic Processes and Smoothing by Splines".
Ann. Inst. Stat. Math. n° 41, 1970. (495 → 502).

/ 7 / P.J. LAURENT
"Approximation et Optimisation".
Hermann, 1972.

/ 8 / F. MERCIER
Thèse Université Paris-Sud (Orsay 06.05.1976).

/ 9 / L. PAIHUA
Thèse Université Scientifique et Médicale de Grenoble (Grenoble 11.05.1978).

/ 10 / M.J.D. POWELL
"Approximation Theory and Methods".
Cambridge University Press, 1981.

/ 11 / C.H. REINSCH
"Smoothing by Spline-Functions".
Num. Math. n° 10, 1967 (177 → 183).

/ 12 / C.H. REINSCH
"Smoothing by Spline Functions II".
Num. Math. n° 16, 1971 (451 → 454).

/ 13 / I.J. SCHOENBERG
Quart. Appl. Math. n° 4, 1946. (45 → 99).

/ 14 / L.L. SCHUMAKER
"Spline Functions : Basic Theory".
Wiley Interscience Publication, 1981.

/ 15 / M. STONE
"Cross Validatory Choice and Assessment of Statistical Prediction".
J. Roy. Stat. Soc. Ser. B 36, 1974 (111 → 147).

/ 16 / R. TEMAM
Cours C_3 Analyse Numérique, Orsay.

/ 17 / F. UTRERAS
Thèse Université Scientifique et Médicale de Grenoble, 1979.

/ 18 / F. UTRERAS
"Quelques résultats d'optimalité pour la méthode de validation croisée".
Séminaire d'Analyse Numérique n° 301.
Grenoble, Nov. 1978.

/ 19 / F. UTREPAS
"Sur le choix du paramètre d'ajustement dans le lissage par fonctions-spline".
Num. Math. n° 34 (tome 1), 1980. (15 → 28).

/ 20 / F. UTREPAS
"Cross Validation Techniques for Smoothing Spline Functions in one or two Dimensions".
Lecture Notes in Math.

/ 21 / G. WAHBA
"Smoothing Noisy Data with Spline Functions".
Num. Math. n° 24, 1975. (383 → 393).

/ 22 / G. WAHBA
"Practical Approximate Solutions to Linear Operator Equations when the Data are Noisy".
Siam. J. Num. Analysis n° 14 (tome 4), 1977.
(651 → 667).

/ 23 / G. WAHBA
"Improprer Priors, Spline Smoothing and the problem of Guarding Against Model Errors in Regression".
J. Roy. Stat. Soc. B. 40, n° 3, 1978. (364 → 372).

/ 24 / G. WAHBA and S. WOLD
"A completly Automatic French curve Fitting Spline Functions by Cross Validation".
Comm. Stat. 4, 1975. (1 - 7).

/ 25 / D.W. MARQUARDT
J. SIAM n° 2 (tome 2), 1963.

A N N E X E I

LE SOUS-PROGRAMME DMAL 2

CISI

M 101

NOVEMBRE 1974

BIBLIOTHEQUE MATHEMATIQUE

NOM AMAL2

OBJET Résolution, par moindres carrés, de système surabondant d'équations linéaires à coefficients réels, avec possibilité de plusieurs seconds membres :

$$\sum_{j=1}^{N} a_{ij}\, x_j^{(k)} = b_i^{(k)} \quad , \quad 1 \leqslant i \leqslant M \ , \ N \leqslant M \ , \ 1 \leqslant k \leqslant L.$$

APPEL CALL AMAL2 (A, IA, M, N, L)

1°- Arguments d'entrée

IA = entier, première dimension du bloc A ci-dessous défini, avec $IA \geqslant M$.

M = entier, nombre d'équations linéaires, avec $M \leqslant 200$.

N = entier, nombre d'inconnues $x_j^{(k)}$, $1 \leqslant j \leqslant N$, avec $2 \leqslant N \leqslant 100$, $N \leqslant M$.

L = entier, nombre de seconds membres, avec $L \geqslant 1$.

A = bloc réel bidimensionné dont la première dimension est IA et la seconde au moins égale à N + L. Il contient les coefficients a_{ij} et les seconds membres $b_i^{(k)}$ du système linéaire :

$$A(i,j) = a_{ij} \quad , \quad A(i, N+k) = b_i^{(k)}$$

pour $1 \leqslant i \leqslant M$, $1 \leqslant j \leqslant N$, $1 \leqslant k \leqslant L$.

2° - Argument de sortie

A = contient, à la place des seconds membres, les L vecteurs inconnus $\{x_i^{(k)}\}$ correspondant aux L seconds membres donnés :

$$A(i, N+k) = x_i^{(k)} \quad , \quad 1 \leqslant i \leqslant N \quad , \quad 1 \leqslant k \leqslant L.$$

La matrice (a_{ij}) et les seconds membres $b_i^{(k)}$ du système étant détruits, il convient de les sauvegarder avant l'appel au sous-programme si besoin est.

Les entiers IA, M, N, L sont inchangés.

Le sous-programme imprime un diagnostic si

- le système linéaire à résoudre est singulier,
- l'une des limitations imposées aux entiers IA, M, N, L n'est pas respectée.

METHODE EMPLOYEE

Pour résoudre le système linéaire surabondant $Ax = b$, A étant une matrice rectangle M x N de rang N, on cherche à déterminer le vecteur x tel que la norme euclidienne :

$$\left(\sum_{i=1}^{M} \left(\sum_{j=1}^{N} a_{ij} x_j - b_i \right)^2 \right)^{1/2}$$

$= \| b - Ax \|$ ou $\| Qb - QAx \|$ soit minimum, Q étant une matrice orthogonale ($Q^t * Q = I$ avec Q^t = transposée de Q, I = matrice unité).

On choisit la matrice Q telle que

$$QA = R = \begin{bmatrix} \tilde{R} \\ O \end{bmatrix} \begin{matrix} \} \ N \times N \\ \} \ (M-N) \times N \end{matrix}$$

où $\tilde{R}$ est une matrice triangulaire supérieure. Cette décomposition est effectuée au moyen des transformations de HOUSEHOLDER.

M 101

REFERENCE

On pourra se reporter à :

- G. GOLUB
Numerical Methods for Solving Linear Least Squares Problems.
Numerisch Mathematik, Vol. 7, n° 3, pp. 206-216, 1965.

EXEMPLE SIMPLE D'APPLICATION

1°- Soit à résoudre, par moindres carrés, un système surabondant de $M = 11$ équations linéaires à $N = 3$ inconnues, avec $L = 3$ seconds membres.

2°- Résolution par le sous-programme AMAL2

```
      DIMENSION  A(20,6) , AA(20,6)
C
      IA = 20
      M = 11
      N = 3
      L = 3
C
C   LECTURE DES COEFFICIENTS DU SYSTEME LINEAIRE
      DO 2 I=1,M
    2 READ (5,4) ( A(I,J),J=1,N )
    4 FORMAT ( 3F6.2 )
C
C   LECTURE DES SECONDS MEMBRES
      DO 6 I=1,M
    6 READ (5,4) ( A(I,K+N),K=1,L )
C
C   CONSERVATION DES COEFFICIENTS ET SECONDS MEMBRES
      NL = N+L
      DO 8 I=1,M
      DO 8 J=1,NL
    8 AA(I,J) = A(I,J)
C
      CALL AMAL2 ( A,IA,M,N,L )
C
C   IMPRESSION DES SOLUTIONS
      DO 10 K=1,L
   10 PRINT 12 , K , ( A(I,K+N),I=1,N )
   12 FORMAT ( // 10H SOLUTION , I3 // ( 6E16.7 ) )
C
      STOP
      END
```

PRINCIPE SUR LEQUEL EST BASE DMAL 2

Soit m et n deux entiers tels que m > n et soit A_n^m une matrice de rang n.

On résoud AX = B en cherchant le vecteur $\hat{X}$ tel que :

$$|| A\hat{X} - B ||_{1^2} = \min_{X} || AX - B ||_{1^2}$$

$$= \min_{X} || QAX - QB ||_{1^2}$$

avec $QQ^T = Q^TQ = I$

On choisit Q de façon que :

$$QA = R = \begin{matrix} \tilde{R} \\ O \end{matrix} \quad \begin{matrix} n \\ m-n \\ n \end{matrix}$$

où $\tilde{R}$ est triangulaire supérieure.

On a alors :

$\hat{X} = \tilde{R}^{-1} \tilde{C}$ où $\tilde{C}$ est le vecteur formé des n premières composantes de QB = C

Par conséquent :

$$B = q^{-1} C$$

et $$A = Q^{-1} R$$

et $$|| A\hat{X} - B ||_{1^2} = || Q^{-1} R\hat{X} - Q^{-1} C ||_{1^2}$$

$$= || C - C ||_{1^2} = \sum_{j=m+1}^{n} C_j^{\,2} \quad ^{1/2}$$

On a donc

$$A = Q^{-1} R$$

c'est-à-dire que

$$A^T = (Q^{-1} R)^T = R^T Q$$

$$A^T A = R^T QQ^{-1} R = R^T R = \tilde{R}^T \tilde{R}$$

(décomposition de Choleski de $A^T A$)

On détermine R par la méthode de Householder.

On pose : $A = A^{(1)}$

et on définit itérativement

$$A^{(K+1)} = p^{(K)} A^{(K)} \qquad K = 1, 2 \ldots n$$

où les β_K sont donnés par :

$$\beta_K = \sigma_K (\sigma_K + |a_{KK}^{(K)}|)^{-1}$$

$$\sigma_K = \left(\sum_{i=1}^{m} a_{iK}^{(K)\ 2} \right)^{1/2}$$

. $i < K$ $\qquad u_i^{(K)} = 0$

. $u_K^{(K)} = \operatorname{sgn}(a_{K,K}^{(K)})\ (\sigma_K + |a_{KK}^{(K)}|)$

. $i > K$ $\qquad u_i^{(K)} = a_{iK}^{(K)}$

Alors :

$P^{(K)}$ est symétrique et orthogonale

(cf. R. TEMAM / 16 /)

on aura en fin de compte

$$R = A^{(n+1)}$$

et $$Q = P^{(n)}\ P^{(n-1)} \ \ldots\ P^{(1)}$$

A N N E X E II

LE SOUS-PROGRAMME DMAGT 1

CISI

M 312

NOVEMBRE 1974

BIBLIOTHEQUE MATHEMATIQUE

NOM AMAGT1

OBJET Ajustement, par moindres carrés, d'une fonction non linéaire $f(x,b)$, $b = (b_1, b_2, \ldots, b_K)$ à une courbe expérimentale (x_i, y_i) affectée des poids p_i, $1 \leqslant i \leqslant N$, avec calcul de la matrice d'erreurs.

On cherche à déterminer les paramètres b minimisant la somme :

$$S(x_i,b) = \sum_{i=1}^{N} p_i \left[y_i - f(x_i,b) \right]^2 .$$

APPEL CALL AMAGT1 (A,IA,B,Y,N,K,EPS,FONC,P)

1°- Arguments d'entrée

IA = entier, première dimension du bloc A ci-dessous défini, avec $IA \geqslant N$.

N = entier, nombre de points expérimentaux, avec $N > K$.

Y,P = blocs réels unidimensionnés de dimension au moins égale à N, contenant respectivement les oordonnées expérimentales et les poids :

$$Y(i) = y_i \quad , \quad P(i) = p_i \quad , \quad 1 \leqslant i \leqslant N .$$

K = entier, nombre de paramètres à ajuster, avec $K \leqslant 100$.

B = bloc réel unidimensionné de dimension au moins égale à K, contenant les valeurs initiales données aux paramètres :

$$B(k) = b_k^{(o)} \quad , \quad 1 \leqslant k \leqslant K \quad .$$

M 312

A = bloc réel bidimensionné dont la première dimension est IA et la seconde au moins égale à K + 2, contenant les valeurs de la fonction f et de ses dérivées partielles premières (jacobien) :

$$A(i, K+2) = f(x_i, b) \quad , \quad 1 \leqslant i \leqslant N$$

$$A(i,j) = \frac{\partial f(x_i, b)}{\partial b_j} = J_{ij}(b) \quad , \quad 1 \leqslant i \leqslant N \quad , \quad 1 \leqslant j \leqslant K \quad .$$

Le tableau A peut ne pas être initialisé avant l'appel au sous-programme AMAGT1. Ce dernier utilise le sous-programme FONC décrit ci-dessous.

FONC=nom d'un sous-programme sans argument, écrit par l'utilisateur, et calculant les valeurs de la fonction f et de son jacobien dans le tableau A ci-dessus mentionné.
Les valeurs nécessaires aux calculs :

- nombre de points N
- nombre de paramètres K
- valeurs des paramètres B
- valeurs des abscisses X à définir,

et le tableau A contenant les résultats seront transmis par un commun blanc ou étiqueté.

On fera par exemple, figurer dans le programme principal et dans le sous-programme FONC, l'instruction :

```
COMMON /TATA/ X,A,B,N,K
```

L'instruction EXTERNAL FONC devra figurer dans le programme appelant.

EPS = scalaire réel contenant l'épsilonn du test de convergence (erreur relative maximum tolérée sur les paramètres b_k) :

$$\frac{|b_k - b'_k|}{\zeta + |b_k|} < \varepsilon \quad , \quad 1 \leqslant k \leqslant K \quad , \quad \zeta = 10^{-5}$$

où b'_k représente la valeur du paramètre à l'itération précédente.

2°- Arguments de sortie

B = contient les paramètres ajustés b_k à la place des paramètres initiaux $b_k^{(o)}$:

$$B(k) = b_k \quad , \quad 1 \leqslant k \leqslant K \quad .$$

A = contient les différences $b_k - b_k'$ calculées aux deux dernières itérations dans la (K + 1)ème colonne :

M 312

$$A(k, K+1) = b_k - b'_k \quad , \quad 1 \leqslant k \leqslant K$$

les ordonnées ajustées dans la (K + 2)ème colonne :

$$A(i, K+2) = f(x_i, b) \quad , \quad 1 \leqslant i \leqslant N$$

et la matrice d'erreurs :

$$A(i,j) = \frac{1}{N-K} \left\{ \sum_{r=1}^{N} p_r \left[y_r - f(x_r, b) \right]^2 \right\} c_{ij} \quad , \quad 1 \leqslant i, j \leqslant K$$

où c_{ij} est le terme général de l'inverse de la matrice $C = J^t * P * J$, avec J = jacobien (N x K) de la fonction f, J^t = transposée de J et P = matrice diagonale (N x N) des poids.

L'approximation initiale $b_k^{(o)}$, $1 \leqslant k \leqslant K$ des paramètres étant détruite, il convient de la sauvegarder avant l'appel au sous-programme si besoin est.

Les entiers IA, N, K, les blocs X, Y, P et le scalaire EPS sont inchangés.

Le sous-programme imprime un diagnostic si :

- le nombre K des paramètres est supérieur à 100.
- la fonction ajustée f est indépendante de l'un des paramètres.

METHODE EMPLOYEE

L'application de la méthode des moindres carrés au modèle linéarisé conduit à la résolution d'un système d'équations linéaires d'ordre K. L'algorithme utilisé est celui de MARQUARDT qui se situe entre la méthode de série de TAYLOR et celle du gradient négatif.

REFERENCES

On pourra se reporter à :

- D.W. MARQUARDT
An Algorithm for Least - Squares Estimation of Nonlinear Parameters.
Journal of the Society for Industrial and Applied Mathematics, Vol. 11, n° 2, pp. 431-441, 1963.

M 312

- H. NGUYEN NGOC
  Minimisation d'une somme de carrés avec calcul du jacobien.
  Rapport CISI n° 465, 1971.

EXEMPLE SIMPLE D'APPLICATION

1°- Soit à ajuster la fonction $f(x,b)$ dépendant de $K = 4$ paramètres :

$$f(x,b) = \frac{b_1 x \ (x + b_2)}{x\,(x + b_3) + b_4}$$

à la courbe expérimentale de $N = 11$ points : (x_i , y_i) , $1 \leqslant i \leqslant N$ ci-après donnée.
Les poids p_i sont égaux à l'unité et l'approximation initiale des paramètres est :

$$b_1^{(o)} = 0.25 \ , \quad b_2^{(o)} = 0.39 \ , \quad b_3^{(o)} = 0.415 \ , \quad b_4^{(o)} = 0.39 \ .$$

Les dérivées partielles premières de f s'écrivent :

$$\frac{\partial f}{\partial b_1} = \frac{x\,(x + b_2)}{x\,(x + b_3) + b_4} \qquad \frac{\partial f}{\partial b_2} = \frac{b_1 \, x}{x\,(x + b_3) + b_4}$$

$$\frac{\partial f}{\partial b_3} = -\frac{b_1 \, x^2 \, (x + b_2)}{\left[x\,(x + b_3) + b_4\right]^2} \qquad \frac{\partial f}{\partial b_4} = -\frac{b_1 \, x\,(x + b_2)}{\left[x\,(x + b_3) + b_4\right]^2}$$

PRINCIPE SUR LEQUEL EST BASE DMAGT 1

Soit $f : {}^{K} \times Q \rightarrow$

$$(b, x_i) \rightarrow f_i(b)$$

On cherche à minimiser pour $Y = (y_i)_{i=1,N}$ donné

$$\phi(b) = \sum_{i=1}^{N} f_i(b) - y_i \; {}^{2}$$

sur l'ensemble des $b = (b_i)$ ^{K}.

Par un développement de Taylor on obtient (développement au 1er ordre) :

$$f_i(b+t) \cong f_i(b) + \sum_{j=1}^{K} \frac{\partial f_i}{\partial b_j}(b) \times t_j \qquad {}^{t} \; {}^{K}$$

Soit, en posant $J = \left(\frac{\partial f_i}{\partial b_j}\right)_{\substack{i=1,N \\ j=1,K}}$ = jacobien de f

$$f_i(b + t) \cong f_i(b) + \sum_{j=1}^{K} J_{ij}(b)\, t_j$$

et donc

$$\phi\ (b + t) = \sum_{i=1}^{N} y_i - f_i\ (b) - \sum_{m=1}^{K} J_{im}\ (b)\ t_j \qquad 2$$

dont le minimum est donné par l'équation

$$\sum_{i=1}^{N} y_i - f_i\ (b) - \sum_{m=1}^{K} J_{im}\ (b)\ t_j \quad J_{ij} = 0$$

$$j = 1 \ldots . K$$

i.e. :

$$\sum_{m=1}^{K} \sum_{i=1}^{N} J_{im}\ (b)\ J_{ij}\ (b) = \sum_{i=1}^{N} (y_i - f_i\ (b))\ J_{ij}\ (b)$$

Soit encore :

$$J^T\ J\ t = J^T\ (y - f)$$

que l'on écrit : $A\ t = g$

D.W. MARQUARTD / 26 / suggère de résoudre:

$$(A + \lambda I)\ \delta = g$$

$\lambda \to 0$ $\quad$ δ = direction de Taylor

$\lambda \to \infty$ $\quad$ $\delta \to$ direction de gradient de descente

ANNEXE III

LISTING DE DÉBRUITAGE

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

   ISN 0002            SUBROUTINE SMOOTH (F,SIGM,SIGM1,VP,N,H,RO)
                     C
   ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
                     C
   ISN 0004            DIMENSION F(1),SIGM(1),SIGM1(1)
   ISN 0005            DIMENSION VP(1)
                     C
   ISN 0006            CALL VALPRO (VP,H,N,F)
                     C
                     C  ETOL=EPSILON DE TOLERANCE POUR L'APPROCHE DU RO MINIMUM
                     C
   ISN 0007            ETOL=1.D-5
                     C
   ISN 0008            CALL SECDOR (F,SIGM,SIGM1,VP,N,H,RO,ETOL)
                     C
   ISN 0009            CALL AJUST2 (F,SIGM,SIGM1,N,H,RO)
                     C
   ISN 0010            RETURN
   ISN 0011            END
```

/ SMOOTH / SIZE OF PROGRAM 00024A HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
F	SFA XR	R*8	000000	H	SFA	R*8	0000D8	N	SFA	I*4	0000DO	RO	SFA	R*8	0000EO
VP	SFA XR	R*8	000000	ETOL	SFA	R*8	0000E8	SIGM	SFA XR	R*8	000000	SIGM1	SFA XR	R*8	000000
AJUST2	SF XF		000000	SECDOR	SF XF		000000	SMOOTH		R*8	0000FO	VALPRO	SF XF		000000

COMPILER GENERATED LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
100001	2	000124									

```
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      10, PROGRAM SIZE =      586, SUBPROGRAM NAME =SMOOTH

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                  186K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

ISN 0002              SUBROUTINE VALPRO (VP,H,N,F)
                    C
                    C
                    C  BUT:
                    C       CALCUL DES VALEURS PROPRES DE LA MATRICE OMEGA
                    C
                    C
ISN 0003              IMPLICIT REAL*8(A-H,O-Z)
                    C
ISN 0004              COMMON /STAT/ FSUP,ALFA
ISN 0005              COMMON /CARTYP/ GMAX,GMIN,GAMA(250)
ISN 0006              DIMENSION VP(1)
ISN 0007              DIMENSION F(1)
                    C
                    C  TABLEAU DES DIX PREMIERES VALEURS PROPRES
ISN 0008              VP(1)=1.5056187314194D0
ISN 0009              VP(2)=2.4997526700739 7D0
ISN 0010              VP(3)=3.5000106794359 1D0
ISN 0011              VP(4)=4.4999995284835 8D0
ISN 0012              VP(5)=5.5000000199439 0D0
ISN 0013              VP(6)=6.4999999913815D0
ISN 0014              VP(7)=7.5000000000372 5D0
ISN 0015              VP(8)=8.4999999999839D0
ISN 0016              VP(9)=9.5000000000007D0
ISN 0017              VP(10)=10.5D0
                    C
ISN 0018              PI=3.141592653589793D0
ISN 0019              CONS=PI/(N*H)
                    C
ISN 0020              DO 1 I=1,10
ISN 0021                 VV=CONS*VP(I)
ISN 0022                 VV=VV*VV
ISN 0023                 VV=VV*VV
ISN 0024                 VP(I)=VV
ISN 0025           1     CONTINUE
                    C
                    C  CALCUL DES AUTRES VALEURS PROPRES
ISN 0026              N2=N-2
ISN 0027              DO 2 I=11,N2
ISN 0028                 VV=(I+5.D-1)*CONS
ISN 0029                 VV=VV*VV
ISN 0030                 VV=VV*VV
ISN 0031                 VP(I)=VV
ISN 0032           2     CONTINUE
                    C
                    C
ISN 0033              S=0.D0
ISN 0034              DO 4 I=1,N
ISN 0035                 IF (F(I).LE.(FSUP*ALFA/100.D0)) GO TO 4
ISN 0037                 A=1.D0/F(I)
ISN 0038                 S=S+A
ISN 0039           4     CONTINUE
                    C
```

```
                C  --------COEFFICIENTS DE POIDS--------
                C
  ISN 0040            GMAX=0.DO
  ISN 0041            GMIN=0.DO
  ISN 0042            DO 5 I=1,N
  ISN 0043               IF (F(I).LE.(FSUP*ALFA/100.DO)) GO TO 5
  ISN 0045               GAMA(I)=1.DO/((F(I)/DFLOAT(N))*S)
  ISN 0046               GMAX=DMAX1(GMAX,GAMA(I))
  ISN 0047               GMIN=DMIN1(GMIN,GAMA(I))
  ISN 0048        5      CONTINUE
                C
                C
  ISN 0049            RETURN
  ISN 0050            END
```

/ VALPRO / SIZE OF PROGRAM 0003EA HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	SF	R*8	000140	F	F XR	R*8	000000	H	F	R*8	000148	I	SFA	I*4	000134
N	FA	I*4	000138	S	SF	R*8	000150	N2	SF	I*4	00013C	PI	SF	R*8	000158
VP	SF XR	R*8	000000	VV	SF	R*8	000160	ALFA	C	R*8	000008	CONS	SF	R*8	000168
FSUP	C	R*8	000000	GAMA	SFA C	R*8	000010	GMAX	SFA C	R*8	000000	GMIN	SFA C	R*8	000008
VALPRO		R*8	000170												

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK * STAT* SIZE OF BLOCK 000010 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
FSUP	R*8	000000	ALFA	R*8	000008						

NAME OF COMMON BLOCK *CARTYP* SIZE OF BLOCK 0007E0 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
GMAX	R*8	000000	GMIN	R*8	000008	GAMA	R*8	000010			

SOURCE STATEMENT LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
1	25	000270	2	32	0002B2	4	39	000304	5	48	000376

COMPILER GENERATED LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
100001	2	0001D4	100002	21	000262	100003	26	000274	100004	28	000290
100005	33	0002B8	100006	35	0002EA	100007	37	0002F6	100008	40	000308
100009	43	000332	100010	45	00033E	100011	49	00037A			

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 49, PROGRAM SIZE = 1002, SUBPROGRAM NAME =VALPRO

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

ISN 0002                SUBROUTINE SECDOR (F,SIGM,SIGM1,VP,N,H,RO,ETOL)
                      C
                      C
                      C  BUT:
                      C      MINIMISATION PAR 'SECTION DOREE'
                      C
                      C
ISN 0003                IMPLICIT REAL*8(A-H,O-Z)
                      C
ISN 0004                DIMENSION F(1),SIGM(1),SIGM1(1)
ISN 0005                DIMENSION VP(1)
                      C
                      C  -----RECHERCHE D'UN ENCADREMENT DU RO MINIMUM-----
                      C
ISN 0006                R1=1.D0
ISN 0007                R2=10.D0
ISN 0008                CALLCROVAL (F,SIGM,SIGM1,VP,N,H,R2,GF2)
ISN 0009           2    CALLCROVAL (F,SIGM,SIGM1,VP,N,H,R1,GF1)
ISN 0010                IF (GF1.GT.GF2) GO TO 1
ISN 0012                R2=R1
ISN 0013                GF2=GF1
ISN 0014                R1=R1/10.D0
ISN 0015                GO TO 2
ISN 0016           1    R3=10.D0*R2
ISN 0017           4    CALLCROVAL (F,SIGM,SIGM1,VP,N,H,R3,GF3)
ISN 0018                IF (GF3.GT.GF2) GO TO 3
ISN 0020                R2=R3
ISN 0021                GF2=GF3
ISN 0022                R3=10.D0*R3
ISN 0023                GO TO 4
                      C-
                      C  --------APPROCHE DU RO MINIMUM---------
                      C
ISN 0024           3    R2=R3
ISN 0025                GF2=GF3
ISN 0026                ITER=0
ISN 0027                TAU=1.8180339890D0
ISN 0028                DELTA=(R2-R1)/TAU
ISN 0029                R4=R1+DELTA
ISN 0030                R3=R2-DELTA
ISN 0031                CALLCROVAL (F,SIGM,SIGM1,VP,N,H,R3,GF3)
ISN 0032                CALLCROVAL (F,SIGM,SIGM1,VP,N,H,R4,GF4)
ISN 0033           6    IF (GF3-GF4) 5,5,7
ISN 0034           5    R2=R4
ISN 0035                GF2=GF4
ISN 0036                R4=R3
ISN 0037                GF4=GF3
ISN 0038                DELTA=DELTA/TAU
ISN 0039                R3=R2-DELTA
ISN 0040                CALLCROVAL (F,SIGM,SIGM1,VP,N,H,R3,GF3)
ISN 0041                ITER=ITER+1
ISN 0042                GO TO 8
ISN 0043           7    R1=R3
```

```
ISN 0044            GF1=GF3
ISN 0045            R3=R4
ISN 0046            GF3=GF4
ISN 0047            DELTA=DELTA/TAU
ISN 0048            R4=R1+DELTA
ISN 0049            CALLCROVAL (F,SIGM,SIGM1,VP,N,H,R4,GF4)
ISN 0050            ITER=ITER+1
                C
                C  --------TEST SUR LA PRECISION--------
                C
ISN 0051        8   ERR=(R2-R1)/(R2+R1)
ISN 0052            IF (ERR.GT.ETOL) GO TO 6
ISN 0054            RO=(R1+R2)/2.DO
                C
ISN 0055            RETURN
ISN 0056            END
```

/ SECDOR / SIZE OF PROGRAM 00049E HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
F	SFA XR	R*8	000000	H	SFA	R*8	000130	N	SFA	I*4	000128	RO	S	R*8	000138
R1	SFA	R*8	000140	R2	SFA	R*8	000148	R3	SFA	R*8	000150	R4	SFA	R*8	000158
VP	SFA XR	R*8	000000	ERR	S	R*8	000160	GF1	SFA	R*8	000168	GF2	SFA	R*8	000170
GF3	SFA	R*8	000178	GF4	SFA	R*8	000180	TAU	SF	R*8	000188	ETOL		R*8	000190
ITER	SF	I*4	00012C	SIGM	SFA XR	R*8	000000	DELTA	SF	R*8	000198	SIGM1	SFA XR	R*8	000000
CROVAL	SF XF		000000	SECDOR		R*8	0001A0								

SOURCE STATEMENT LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
2	9	000202	1	16	000240	4	17	00024C	3	24	000282
6	33	0002DC	5	34	0002E8	7	43	00032E	8	51	000370

COMPILER GENERATED LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
100001	2	0001CC	100002	12	000220	100003	20	00026A	100004	54	00038A

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 55, PROGRAM SIZE = 1182, SUBPROGRAM NAME =SECDOR

STATISTICS NO DIAGNOSTICS GENERATED

****** END OF COMPILATION ****** 186K BYTES OF CORE NOT USED

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

ISN 0002            SUBROUTINE CROVAL (F,SIGM,SIGM1,VP,N,H,RO,CV)
                    C
                    C
                    C  BUT:
                    C      CALCUL DE LA FONCTION DE VALIDATION CROISEE
                    C
                    C
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
                    C
ISN 0004            COMMON /CARTYP/ GMAX,GMIN,GAMA(250)
ISN 0005            DIMENSION F(1),SIGM(1),SIGM1(1)
ISN 0006            DIMENSION VP(1)
                    C
ISN 0007            CALL AJUST2(F,SIGM,SIGM1,N,H,RO)
                    C
ISN 0008            SN=0.DO
ISN 0009            DO 3 I=1,N
ISN 0010               SN=SN+(F(I)-SIGM(I))**2*GAMA(I)
ISN 0011         3  CONTINUE
                    C
ISN 0012            SN=SN/DFLOAT(N)
ISN 0013            TAR=2.DO
ISN 0014            N2=N-2
                    C
ISN 0015            DO 4I=1,N2
ISN 0016               C=VP(I)/RO
ISN 0017               C=1.DO/(1.DO+C/DFLOAT(N))
ISN 0018               TAR=TAR+C
ISN 0019         4  CONTINUE
                    C
ISN 0020            CV=1.DO-TAR/DFLOAT(N)
ISN 0021            CV=CV*CV
ISN 0022            IF (CV.LE.1.D-16) GO TO 5
ISN 0024            CV=SN/CV
                    C
ISN 0025            WRITE (6,60) RO,CV,GMAX,GMIN
ISN 0026        60  FORMAT (10X,'RO=',E7.1,10X'CV=',E12.6,10X,'GMAX=',E12.6,10X,'GMIN=
                   &',E12.6)
                    C
ISN 0027            RETURN
ISN 0028         5  CV=1.D25
ISN 0029            WRITE (6,60) RO,CV,GMAX,GMIN
ISN 0030            RETURN
ISN 0031            END
```

/ CROVAL / SIZE OF PROGRAM 0003D0 HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
C	SF	R*8	000118	F	SFA XR	R*8	000000	H	SFA	R*8	000120	I	SF	I*4	000108
N	SFA	I*4	00010C	CV	SF	R*8	000128	N2	SF	I*4	000110	RO	SFA	R*8	000130
SN	SF	R*8	000138	VP	F XR	R*8	000000	TAR	SF	R*8	000140	GAMA	F C	R*8	000010
GMAX	F C	R*8	000000	GMIN	F C	R*8	000008	SIGM	SFA XR	R*8	000000	SIGM1	SFA XR	R*8	000000

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

   ISN 0002            SUBROUTINE AJUST2 (F,SIGM,SIGM1,N,H,RO)
                     C
                     C
                     C  BUT:
                     C       CALCUL DE LA VALEUR AUX NOEUDS DE LA SPLINE D'AJUSTEMENT SUR LES
                     C       DONNEES EXPERIMENTALES POUR UN RO DONNE
                     C
                     C
   ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
                     C
   ISN 0004            DIMENSION F(1),SIGM(1),SIGM1(1)
   ISN 0005            DIMENSION A(221,2,2)
                     C
                     C
                     C --------SAUVEGARDE DE LA FONCTION EXPERIMENTALE F--------
                     C
   ISN 0006            DO 1 I=1,N
   ISN 0007               SIGM(I)=F(I)
   ISN 0008          1 CONTINUE
                     C
                     C
   ISN 0009            ALFA=H*H*H*RO/6.DO
   ISN 0010            DET=1.DO+ALFA+ALFA
   ISN 0011            Z1=ALFA*SIGM(1)/DET
   ISN 0012            BETA=4.DO+ALFA
   ISN 0013            SIGM(1)=2.DO*Z1
   ISN 0014            SIGM1(1)=-3.DO*Z1
   ISN 0015            A11=1.DO/DET
   ISN 0016            A(1,1,1)=-A11
   ISN 0017            A(1,1,2)=A11
   ISN 0018            A(1,2,1)=-3.DO*ALFA/DET
   ISN 0019            A(1,2,2)=(ALFA-1.DO)/DET
                     C
                     C  --------ELIMINATION PAR BLOCS--------
                     C
   ISN 0020            N1=N-1
                     C
   ISN 0021            DO 2 I=2,N1
   ISN 0022               A1=2.DO*A(I-1,1,1)+A(I-1,2,1)
   ISN 0023               A2=2.DO*A(I-1,1,2)+A(I-1,2,2)
   ISN 0024               B1=2.DO*SIGM(I-1)+SIGM1(I-1)
   ISN 0025               A22=BETA+A1
   ISN 0026               A12=A2
   ISN 0027               A21=-A1-A(I-1,1,1)
   ISN 0028               A11=4.DO-A2-A(I-1,1,2)
   ISN 0029               Z1=ALFA*SIGM(I)+B1
   ISN 0030               D1=-B1-SIGM(I-1)
   ISN 0031               DET=A11*A22-A12*A21
   ISN 0032               A11=A11/DET
   ISN 0033               A12=-A12/DET
   ISN 0034               A21=-A21/DET
   ISN 0035               A22=A22/DET
   ISN 0036               A(I,1,1)=-2.DO*A11-3.DO*A12
```

```
ISN 0037                A(I,1,2)=A11+A12
ISN 0038                A(I,2,1)=-2.DO*A21-3.DO*A22
ISN 0039                A(I,2,2)=A21+A22
ISN 0040                SIGM(I)=A11*Z1+A12*D1
ISN 0041                SIGM1(I)=A21*Z1+A22*D1
ISN 0042      2      CONTINUE
                     C
ISN 0043             A1=2.DO*A(N1,1,1)+A(N1,2,1)
ISN 0044             A2=2.DO*A(N1,1,2)+A(N1,2,2)
ISN 0045             B1=2.DO*SIGM(N1)+SIGM1(N1)
ISN 0046             Z1=ALFA*SIGM(N)+B1
ISN 0047             A22=ALFA+2.DO+A1
ISN 0048             A12=-1.DO+A2
ISN 0049             A21=-3.DO-A1-A(N1,1,1)
ISN 0050             A11=2.DO-A2-A(N1,1,2)
ISN 0051             D1=-B1-SIGM(N1)
                     C
                     C ----RESOLUTION DU SYSTEME TRIANGULAIRE SUP. PAR BLOCS----
                     C
ISN 0052             DET=A11*A22-A12*A21
ISN 0053             A11=A11/DET
ISN 0054             A12=-A12/DET
ISN 0055             A21=-A21/DET
ISN 0056             A22=A22/DET
ISN 0057             A(N,1,1)=A11
ISN 0058             A(N,1,2)=A12
ISN 0059             A(N,2,1)=A21
ISN 0060             A(N,2,2)=A22
ISN 0061             SIGM(N)=A11*Z1+A12*D1
ISN 0062             SIGM1(N)=A21*Z1+A22*D1
                     C
ISN 0063             DO 3 J=1,N1
ISN 0064                I=N-J
ISN 0065                SIGM(I)=SIGM(I)-(A(I,1,1)*SIGM(I+1)+A(I,1,2)*SIGM1(I+1))
ISN 0066                SIGM1(I)=SIGM1(I)-(A(I,2,1)*SIGM(I+1)+A(I,2,2)*SIGM1(I+1))
ISN 0067      3      CONTINUE
                     C
ISN 0068             DO 4 I=1,N
ISN 0069                SIGM1(I)=SIGM1(I)/H
ISN 0070      4      CONTINUE
ISN 0071             RETURN
ISN 0072             END
```

/ AJUST2 / SIZE OF PROGRAM 00222C HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A SF		R*8	000140	F F	XR	R*8	000000	H F		R*8	0000C8	I SF		I*4	0000B8
J SF		I*4	0000BC	N F		I*4	0000C0	A1 SF		R*8	0000D0	A2 SF		R*8	0000D8
B1 SF		R*8	0000E0	D1 SF		R*8	0000E8	N1 SF		I*4	0000C4	RO F		R*8	0000F0
Z1 SF		R*8	0000F8	A11 SF		R*8	000100	A12 SF		R*8	000108	A21 SF		R*8	000110
A22 SF		R*8	000118	DET SF		R*8	000120	ALFA SF		R*8	000128	BETA SF		R*8	000130
SIGM SF	XR	R*8	000000	SIGM1 SF	XR	R*8	000000	AJUST2		R*8	000138				

SOURCE STATEMENT LABELS

A N N E X E IV

LISTING DE DÉCONVOLUTION INSTRUMENTALE

REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```
ISN 0002            IMPLICIT REAL*8(A-H,O-Z)
ISN 0003            COMMON /KONST/ K,KREEL,KINS,KDIV,N,NN
ISN 0004            DIMENSION U(250),V(250),G(250),F(250),FBR(250)
ISN 0005            DIMENSION C(250),B(250)
ISN 0006            DIMENSION SIGM(250),SIGM1(250),VP(250)
ISN 0007            DIMENSION X(250)
ISN 0008            DIMENSION GS(250)
ISN 0009            DIMENSION S(250)
ISN 0010            DIMENSION US(250),FS(250),T(250)
                C
                C  LES TITRES DES GRAPHIQUES
                C  ==========================
                C
ISN 0011            DATA TGR1 /8HRESULTAT/
ISN 0012            DATA TGR2 /8HCONVOLEE/
ISN 0013            DATA TGR3 /8HLISSAGE /
                C
                C  INITIALISATIONS A ZERO
                C  ======================
                C
ISN 0014            DO 1000 I=1,250
ISN 0015               U(I)=0.D0
ISN 0016               V(I)=0.D0
ISN 0017               G(I)=0.D0
ISN 0018               F(I)=0.D0
ISN 0019               C(I)=0.D0
ISN 0020               B(I)=0.D0
ISN 0021               X(I)=0.D0
ISN 0022               S(I)=0.D0
ISN 0023               T(I)=0.D0
ISN 0024               VP(I)=0.D0
ISN 0025               GS(I)=0.D0
ISN 0026               US(I)=0.D0
ISN 0027               FS(I)=0.D0
ISN 0028               FBR(I)=0.D0
ISN 0029               SIGM(I)=0.D0
ISN 0030               SIGM1(I)=0.D0
ISN 0031     1000   CONTINUE
                C
                C
ISN 0032            H=1.D0
ISN 0033            KINS=9
ISN 0034            KREEL=201
ISN 0035            IDIV=2
ISN 0036            KDIV=10
ISN 0037            N=221
ISN 0038            K=((N+KDIV-1)/KDIV)-2
ISN 0039            K2=K+2
ISN 0040            NN=(N+IDIV-1)/IDIV
                C
                C  CALCUL DES ABSCISSES
                C  ====================
                C
```

```
ISN 0041            DO 1 I=1,N
ISN 0042               X(I)=DFLOAT(I-1)*H
ISN 0043      1     CONTINUE
              C
              C  CONSTRUCTION DES FONCTIONS-TEST MATHEMATIQUES
              C  ============================================
              C
ISN 0044            CALL INIDO (U,G,F,X,H)
              C
              C  PURGE DE LA MEMOIRE
              C  ====================
              C
ISN 0045            CALL PURGE (20)
              C
              C  BRUITAGE DE LA COURBE TEST
              C  ==========================
              C
ISN 0046            IRAND=575757
ISN 0047            CALL BRUIT (IRAND,F,FBR)
              C
              C  DEBRUITAGE DE LA COURBE 'EXPERIMENTALE'
              C  ======================================
              C
ISN 0048            CALL SMOOTH (FBR,SIGM,SIGM1,VP,N,H,RO)
              C
              C  RECHERCHE DES PARAMETRES DE LA SPLINE MODELISANT LA FONCTION TEST
              C  ================================================================
              C  'INSTRUMENTALE'
              C  ===============
              C
ISN 0049            CALL INSTRU(C,G,GS,H)
ISN 0050            WRITE (6,63)
ISN 0051            WRITE (6,60)
ISN 0052            WRIIE (6,61) (F(I),FBR(I),SIGM(I),F(I),I=1,N)
ISN 0053            WRITE (6,63)
ISN 0054      60    FORMAT (12X,'F(I)',12X,'FBR(I)',12X,'SIGM(I)',12X,'F(I)',/)
              C
              C  RECHERCHE DES PARAMETRES DE LA SPLINE MODELISANT LA FONCTION-TEST
              C  ================================================================
              C  'INFORMATION CHERCHEE'
              C  ======================
              C
ISN 0055            CALL DECONV (C,B,SIGM,H)
              C
              C  CONSTRUCTION DE LA SPLINE V MODELISANT L'INFORMATION CHERCHEE
              C  ============================================================
              C
ISN 0056            CALL RESULT (B,U,V,X,H)
              C
              C  CALCUL (ANALYTIQUE) DE LA CONVOLEE DE LA SPLINE MODELISANT
              C  =========================================================
              C  L'INFORMATON CHERCHEE PAR LA SPLINE MODELISANT LA FONCTION INSTRUMENT
              C  ===================================================================
              C
ISN 0057            CALL RECONV (B,C,S)
              C
              C  CALCUL DE LA MEME CONVOLEE PAR LA FORMULE DES TRAPEZES
              C  ====================================================
```

```
                C
ISN 0058              T(1)=0.DO
ISN 0059              N1=N-1
ISN 0060              DO 2 I=1,N1
ISN 0061                 DO 3 J=1,I
ISN 0062                    T1=V(J)/3.DO+V(J+1)/6.DO
ISN 0063                    T2=V(J)/6.DO+V(J+1)/3.DO
ISN 0064                    T3=GS(I-J+2)*T1+GS(I-J+1)*T2
ISN 0065                    T(I+1)=T(I+1)+H*T3
ISN 0066        3        CONTINUE
ISN 0067        2     CONTINUE
                C
                C  IMPRESSIONS DE SORTIE
                C  ====================
                C
ISN 0068              WRITE (6,63)
ISN 0069        63    FORMAT (///)
ISN 0070              WRITE (6,65)
ISN 0071        65    FORMAT (50X,'COMPARAISON DES CONVOLEES',/,50X,25(1H*),//)
ISN 0072              WRITE (6,61) (F(I),FS(I),S(I),T(I),I=1,N)
ISN 0073        61    FORMAT ((4(12X,E14.7)))
                C
                C  TRACES DE COURBES
                C  =================
                C
ISN 0074              CALL COURBE (FBR,SIGM,X,N,TGR3)
ISN 0075              CALL COURBE (U,G,X,N,TGR1)
ISN 0076              CALL COURBE (F,S,X,N,TGR2)
ISN 0077              CALL COURBE (F,T,X,N,TGR2)
                C
ISN 0078              STOP
ISN 0079              END
```

/ MAIN / SIZE OF PROGRAM 008390 HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
B	SFA	R*8	000268	C	SFA	R*8	000A38	F	SFA	R*8	001208	G	SFA	R*8	0019D8
H	SFA	R*8	000228	I	SFA	I*4	00020C	J	SF	I*4	000210	K	SF C	I*4	000000
N	SFA C	I*4	000010	S	SFA	R*8	0021A8	T	SFA	R*8	002978	U	SFA	R*8	003148
V	SFA	R*8	003918	X	SFA	R*8	0040E8	FS	SF	R*8	0048B8	GS	SFA	R*8	005088
K2	S	I*4	000214	NN	S C	I*4	000014	N1	SF	I*4	000218	RO	SFA	R*8	000230
T1	SF	R*8	000238	T2	SF	R*8	000240	T3	SF	R*8	000248	US	S	R*8	005858
VP	SFA	R*8	006028	FBR	SFA	R*8	0067F8	IDIV	SF	I*4	00021C	KDIV	SF C	I*4	00000C
KINS	S C	I*4	000008	SIGM	SFA	R*8	006FC8	TGR1	SFA	R*8	000250	TGR2	SFA	R*8	000258
TGR3	SFA	R*8	000260	BRUIT	SF XF		000000	INIDO	SF XF		000000	IRAND	SFA	I*4	000220
KREEL	S C	I*4	000004	PURGE	SF XF		000000	SIGM1	SFA	R*8	007798	COURBE	SF XF		000000
DECONV	SF XF		000000	IBCOM#	F XF	R*8	000000	INSTRU	SF XF		000000	RECONV	SF XF		000000
RESULT	SF XF		000000	SMOOTH	SF XF		000000								

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK * KONST* SIZE OF BLOCK 000018 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
K	I*4	000000	KREEL	I*4	000004	KINS	I*4	000008	KDIV	I*4	00000C
N	I*4	000010	NN	I*4	000014						

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

   ISN 0002            SUBROUTINE INIDO (U,G,F,X,H)
                     C
   ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
                     C
   ISN 0004            COMMON /STAT/ FSUP
   ISN 0005            COMMON /KONST/ K,KREEL,KINS,KDIV,N,NN
   ISN 0006            DIMENSION U(1),G(1),F(1)
   ISN 0007            DIMENSION X(1)
   ISN 0008            DIMENSION Y(1000)
                     C
   ISN 0009            KR1=KREEL+1
   ISN 0010            KR2=KREEL+2
   ISN 0011            KI2=KINS+2
   ISN 0012            KI3=KINS+3
                     C
   ISN 0013            ALFA=5.DO
   ISN 0014            BETA=20.DO
   ISN 0015            DELTA=10.DO
   ISN 0016            FSUP=0.DO
                     C
   ISN 0017             PI=2.DO*DARSIN(1.DO)
   ISN 0018             H=PI/40.DO
                     C
   ISN 0019            DO 70 I=1,1000
   ISN 0020               Y(I)=DFLOAT(I-1)*H
   ISN 0021         70 CONTINUE
                     C
   ISN 0022            DO 10 I=1,81
   ISN 0023               U(I)=(1.DO-DCOS(Y(I)))*ALFA
   ISN 0024         10 CONTINUE
   ISN 0025            DO 20 I=61,141
   ISN 0026               U(I)=U(I)+(1.DO+DSIN(Y(I)))*BETA
   ISN 0027         20 CONTINUE
   ISN 0028            DO 30 I=121,201
   ISN 0029               U(I)=U(I)+(1.DO+DCOS(Y(I)))*DELTA
   ISN 0030         30 CONTINUE
                     C
   ISN 0031            VMOYEN=5.DO
   ISN 0032            VSIGMA=DSQRT(5.DO)
                     C
   ISN 0033            DO 100 I=1,KI2
   ISN 0034               G(I)=((X(I)-VMOYEN)/VSIGMA)**2
   ISN 0035               G(I)=-G(I)/2.DO
   ISN 0036               G(I)=DEXP(G(I))
   ISN 0037               G(I)=G(I)/VSIGMA
   ISN 0038               G(I)=G(I)/DSQRT(2.DO*PI)
   ISN 0039               G(I)=G(I)*BETA
   ISN 0040        100 CONTINUE
                     C
   ISN 0041            DO 200 I=1,KR1
   ISN 0042               J=KR2-I+1
   ISN 0043               G(J)=G(J-1)
   ISN 0044        200 CONTINUE
```

```
            C
ISN 0045          DO 250 I=1,211
ISN 0046             J=211-I+11
ISN 0047             U(J)=U(J-10)
ISN 0048             G(J)=G(J-10)
ISN 0049     250  CONTINUE
            C
ISN 0050          DO 260 I=1,10
ISN 0051             U(I)=0.D0
ISN 0052             G(I)=0.D0
ISN 0053     260  CONTINUE
            C
            C  CALCUL DE LA CONVOLEE DE U ET DE G PAR LA METHODE DES TRAPEZES
            C  ==========================================================
            C
ISN 0054          H=1.D0
ISN 0055          N1=N-1
ISN 0056          DO 2 I=1,N1
ISN 0057             DO 3 J=1,I
ISN 0058                F1=U(J)/3.D0+U(J+1)/6.D0
ISN 0059                F2=U(J)/6.D0+U(J+1)/3.D0
ISN 0060                F3=G(I-J+2)*F1+G(I-J+1)*F2
ISN 0061                F(I+1)=F(I+1)+H*F3
ISN 0062     3          CONTINUE
ISN 0063                FSUP=DMAX1(F(I+1),FSUP)
ISN 0064     2       CONTINUE
            C
ISN 0065          RETURN
ISN 0066          END
```

/ INIDO / SIZE OF PROGRAM 002552 HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
F	SFA XR	R*8	000000	G	SFA XR	R*8	000000	H	SF	R*8	000138	I	SFA	I*4	000118
J	SF	I*4	00011C	K	C	I*4	NR	N	F C	I*4	000010	U	SF XR	R*8	000000
X	F XR	R*8	000000	Y	SFA	R*8	000188	F1	SF	R*8	000140	F2	SF	R*8	000148
F3	SF	R*8	000150	NN	C	I*4	NR	N1	SF	I*4	000120	PI	SFA	R*8	000158
KI2	SF	I*4	000124	KI3	S	I*4	000128	KR1	SF	I*4	00012C	KR2	SF	I*4	000130
ALFA	SF	R*8	000160	BETA	SF	R*8	000168	DCOS	F XF	R*8	000000	DEXP	F XF	R*8	000000
DSIN	F XF	R*8	000000	FSUP	SFA C	R*8	000000	KDIV	C	I*4	NR	KINS	F C	I*4	000008
DELTA	SF	R*8	000170	DSQRT	F XF	R*8	000000	INIDO		I*4	000134	KREEL	F C	I*4	000004
DARSIN	F XF	R*8	000000	VMOYEN	SF	R*8	000178	VSIGMA	SF	R*8	000180				

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK * STAT* SIZE OF BLOCK 000008 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
FSUP	R*8	000000									

NAME OF COMMON BLOCK * KONST* SIZE OF BLOCK 000018 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
K	I*4	000000 NR	KREEL	I*4	000004	KINS	I*4	000008	KDIV	I*4	00000C NR
N	I*4	000010	NN	I*4	000014 NR						

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

   ISN 0002            SUBROUTINE PURGE (N)
                     C
                     C
   ISN 0003            IRAND=575757
   ISN 0004            VMOYEN=0.
   ISN 0005            VSIGMA=1.
   ISN 0006            DO 1 I=1,N
   ISN 0007               A=APHERF(IRAND,VMOYEN,VSIGMA)
   ISN 0008      1     CONTINUE
   ISN 0009            RETURN
   ISN 0010            END


                                                    /   PURGE /     SIZE OF PROGRAM 00011A HEXADECIMAL BYTES

      NAME     TAG   TYPE  ADD.        NAME     TAG   TYPE  ADD.        NAME     TAG   TYPE  ADD.        NAME     TAG   TYPE  ADD
         A S         R*4   000098           I SF        I*4   00009C           N  F        I*4   0000A0      IRAND SFA       I*4   0000A4
     PURGE           R*4   0000A8     APHERF  F  XF     R*4   000000     VMOYEN SFA       R*4   0000AC      VSIGMA SFA      R*4   0000B0


  SOURCE STATEMENT LABELS

     LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR
         1    8  0000EC

  COMPILER GENERATED LABELS

     LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR
    100001    2  0000BC       100002    7  0000DE       100003    9  0000F0

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*      SOURCE STATEMENTS =        9, PROGRAM SIZE =       282, SUBPROGRAM NAME = PURGE

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                  186K BYTES OF CORE NOT USED
```

REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```
ISN 0002            SUBROUTINE BRUIT (IRAND,F,FBR)
ISN 0003            COMMON /KONST/ K,KREEL,KINS,KDIV,N,NN
ISN 0004            DOUBLE PRECISION F(1),FBR(1)
              C
ISN 0005            DOUBLE PRECISION H
              C
              C
ISN 0006            DO 1 I=1,N
ISN 0007                  VMOYEN=F(I)
ISN 0008                  VMOYEN=ABS(VMOYEN)
ISN 0009                  VSIGMA=SQRT(VMOYEN)
ISN 0010                  FBR(I)=APHERF(IRAND,VMOYEN,VSIGMA)
ISN 0011       1    CONTINUE
              C
              C
ISN 0012            RETURN
ISN 0013            END
```

/ BRUIT / SIZE OF PROGRAM 00019E HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
F	F XR	R*8	000000	H		R*8	NR	I	SF	I*4	000094	K	C	I*4	NR
N	F C	I*4	000010	NN	C	I*4	NR	FBR	S XR	R*8	000000	KDIV	C	I*4	NR
KINS	C	I*4	NR	SQRT	F XF	R*4	000000	BRUIT		R*4	000098	IRAND	FA	I*4	00009C
KREEL	C	I*4	NR	APHERF	F XF	R*4	000000	VMOYEN	SFA	R*4	0000A0	VSIGMA	SFA	R*4	0000A4

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK * KONST* SIZE OF BLOCK 000018 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
K	I*4	000000 NR	KREEL	I*4	000004 NR	KINS	I*4	000008 NR	KDIV	I*4	00000C NR
N	I*4	000010	NN	I*4	000014 NR						

SOURCE STATEMENT LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
1	11	000132									

COMPILER GENERATED LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
100001	2	0000E0	100002	7	0000FA	100003	12	000136			

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```
ISN 0002            SUBROUTINE INSTRU (C,G,GS,H)
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
ISN 0004            COMMON /KONST/ K,KREEL,KINS,KDIV,N,NN
ISN 0005            DIMENSION C(1),G(1)
ISN 0006            DIMENSION A(221,222)
ISN 0007            DIMENSION GS(1)
                C
                C
ISN 0008            DO 1000 I=1,221
ISN 0009               DO 1001 J=1,222
ISN 0010                  A(I,J)=0.D0
ISN 0011    1001       CONTINUE
ISN 0012    1000    CONTINUE
                C
ISN 0013            KR1=KREEL-1
ISN 0014            KR2=KREEL-2
ISN 0015            KI2=KINS+2
ISN 0016            N1=N-1
ISN 0017            N2=N-2
                C
ISN 0018            DO 1 I=1,N
ISN 0019               A(I,1)=1.D0
ISN 0020               A(I,2)=DFLOAT(I-1)*H
ISN 0021               IF (I.LT.3) GO TO 1
ISN 0023               DO 2 J=3,I
ISN 0024                  A(I,J)=((DFLOAT (I-J+1)*H)**3)/6.D0
ISN 0025    2          CONTINUE
ISN 0026    1       CONTINUE
                C
ISN 0027            DO 50 I=1,N2
ISN 0028               DO 51 J=1,N
ISN 0029                  A(I,J)=A(I+1,J)
ISN 0030    51         CONTINUE
ISN 0031    50      CONTINUE
                C
ISN 0032            DO 52 J=3,N
ISN 0033               A(N1,J)=1.D0
ISN 0034               A(N,J)=DFLOAT(J-2)*H
ISN 0035    52      CONTINUE
                C
                C
ISN 0036            DO 53 I=1,N2
ISN 0037               A(I,N+1)=G(I+1)
ISN 0038    53      CONTINUE
                C
ISN 0039            DO 54 I=N1,N
ISN 0040               A(I,1)=0.D0
ISN 0041               A(I,2)=0.D0
ISN 0042               A(I,N+1)=0.D0
ISN 0043    54      CONTINUE
                C
                C
ISN 0044            M=N
```

```
                C
                C  TRIANGULATION DE LA MATRICE A
                C
ISN 0045              DO 10 L=2,M
ISN 0046                 I=M-L+1
ISN 0047                 I1=I+1
ISN 0048                 ALFA=A(I,I1)/A(I1,I1)
ISN 0049                 A(I,M+1)=A(I,M+1)-A(I1,M+1)*ALFA
ISN 0050                 DO 11 J=1,I1
ISN 0051                    A(I,J)=A(I,J)-A(I1,J)*ALFA
ISN 0052        11       CONTINUE
ISN 0053        10    CONTINUE
                C
                C  RESOLUTION DU SYSTEME TRIANGULAIRE
                C
ISN 0054              C(1)=A(1,M+1)/A(1,1)
ISN 0055              DO 20 I=2,M
ISN 0056                 C(I)=A(I,M+1)
ISN 0057                 I1=I-1
ISN 0058                 DO 21 J=1,I1
ISN 0059                    C(I)=C(I)-A(I,J)*C(J)
ISN 0060        21       CONTINUE
ISN 0061                 C(I)=C(I)/A(I,I)
ISN 0062        20    CONTINUE
                C
                C
                C
ISN 0063              DO 5 I=1,N
ISN 0064                 GS(I)=C(1)
ISN 0065                 GS(I)=GS(I)+C(2)*DFLOAT(I-1)*H
ISN 0066                 DO 6 J=1,N2
ISN 0067                    IJ=MAX(I-1-J,0)
ISN 0068                    E=DFLOAT(IJ)*H
ISN 0069                    E=E**3/6.D0
ISN 0070                    GS(I)=GS(I)+C(J+2)*E
ISN 0071        6        CONTINUE
ISN 0072        5     CONTINUE
                C
ISN 0073              WRITE (6,64)
ISN 0074        64    FORMAT (1H1)
ISN 0075              WRITE (6,62) (G(I),GS(I),C(I),I=1,N)
ISN 0076        62    FORMAT (10X,E14.7,10X,E14.7,30X,E14.7)
ISN 0077        63    FORMAT (//)
ISN 0078              WRITE (6,63)
ISN 0079              WRITE (6,63)
                C
ISN 0080              RETURN
ISN 0081              END
```

/ INSTRU / SIZE OF PROGRAM 060508 HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	SF	R*8	000130	C	SF XR	R*8	000000	E	SF	R*8	000118	G	F XR	R*8	000000
H	F	R*8	000120	I	SFA	I*4	0000E4	J	SFA	I*4	0000E8	K	C	I*4	NR
L	SF	I*4	0000EC	M	SF	I*4	0000F0	N	F C	I*4	000010	GS	SF XR	R*8	000000
IJ	SFA	I*4	0000F4	I1	SF	I*4	0000F8	NN	C	I*4	NR	N1	SF	I*4	0000FC
N2	SF	I*4	000100	KI2	S	I*4	000104	KR1	S	I*4	000108	KR2	S	I*4	00010C

REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```
ISN 0002            SUBROUTINE DECONV (C,B,SIGM,H)
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
ISN 0004            COMMON /KONST/ K,KREEL,KINS,KDIV,N,NN
ISN 0005            DIMENSION C(1),B(1),SIGM(1)
ISN 0006            DIMENSION RC(220),RQ(220),TC(31),TQ(31),TRS(31,219)
ISN 0007            DIMENSION A(70,34)
                C
                C
ISN 0008            DO 1000 I=1,70
ISN 0009               DO 1001 J=1,32
ISN 0010                  A(I,J)=0.D0
ISN 0011      1001     CONTINUE
ISN 0012      1000  CONTINUE
                C
ISN 0013            K1=K+1
ISN 0014            KK=K+9
ISN 0015            KK2=KK+2
ISN 0016            NM=0
ISN 0017            KR2=KREEL-2
ISN 0018            K2=K+2
ISN 0019            N1=N-1
ISN 0020            N2=N-2
                C
ISN 0021            HQ=H**4/24.D0
ISN 0022            HC=H**5/120.D0
ISN 0023            HS=H**7/5040.D0
                C
ISN 0024            DO 1 I=6,221,5
ISN 0025               NM=NM+1
ISN 0026               I1=I-1
ISN 0027               DO 2 L=1,N2
ISN 0028                  IL=MAX(I1-L,0)
ISN 0029                  RQ(L)=(DFLOAT(IL)**4)*HQ
ISN 0030                  RC(L)=(DFLOAT(IL)**5)*HC
ISN 0031      2        CONTINUE
ISN 0032               DO 3 L=1,K
ISN 0033                  IL=MAX(I1-10*L,0)
ISN 0034                  TQ(L)=(DFLOAT(IL)**4)*HQ
ISN 0035                  TC(L)=(DFLOAT(IL)**5)*HC
ISN 0036                  DO 4 M=1,N2
ISN 0037                     ILM=MAX(I1-10*L-M,0)
ISN 0038                     TRS(L,M)=(DFLOAT(ILM)**7)*HS
ISN 0039      4           CONTINUE
ISN 0040      3        CONTINUE
ISN 0041               DO 33 L=K1,KK
ISN 0042                  IL=MAX(I1-210-(L-21),0)
ISN 0043                  TQ(L)=(DFLOAT(IL)**4)*HQ
ISN 0044                  TC(L)=(DFLOAT(IL)**5)*HC
ISN 0045                  DO 44 M=1,N2
ISN 0046                     ILM=MAX(I1-210-(L-21)-M,0)
ISN 0047                     TRS(L,M)=(DFLOAT(ILM)**7)*HS
ISN 0048      44          CONTINUE
ISN 0049      33       CONTINUE
```

```
            C
ISN 0050              X=I1*H
ISN 0051              X2=X*X/2.DO
ISN 0052              X3=X2*X/3.DO
            C
ISN 0053              A(NM,1)=C(1)*X+C(2)*X2
ISN 0054              A(NM,2)=C(1)*X2+C(2)*X3
ISN 0055              DO 5 M=1,N2
ISN 0056                 A(NM,1)=A(NM,1)+C(2+M)*RQ(M)
ISN 0057                 A(NM,2)=A(NM,2)+C(2+M)*RC(M)
ISN 0058     5        CONTINUE
ISN 0059              DO 6 J=3,KK2
ISN 0060                 A(NM,J)=C(1)*TQ(J-2)+C(2)*TC(J-2)
ISN 0061                 DO 7 M=1,N2
ISN 0062                    A(NM,J)=A(NM,J)+C(2+M)*TRS(J-2,M)
ISN 0063     7           CONTINUE
ISN 0064     6        CONTINUE
ISN 0065              A(NM,KK+3)=SIGM(I)
ISN 0066     1     CONTINUE
            C
ISN 0067           A(1,KK+3)=0.DO
ISN 0068           A(2,KK+3)=0.DO
            C
            C
ISN 0069           DO 14 I=211,220
ISN 0070              NM=NM+1
ISN 0071              A(NM,1)=1.DO
ISN 0072              A(NM,2)=DFLOAT(I-1)*H
ISN 0073              DO 15 J=3,23
ISN 0074                 A(NM,J)=((DFLOAT(I-1-10*(J-2))*H)**3)/6.DO
ISN 0075     15       CONTINUE
ISN 0076              IF (I.LT.213) GO TO 14
ISN 0078              IJ=I+22-211
ISN 0079              DO 16 J=24,IJ
ISN 0080                 A(NM,J)=((DFLOAT(I-1-210-(J-2-21))*H)**3)/6.DO
ISN 0081     16       CONTINUE
ISN 0082              A(NM,KK+3)=0.DO
ISN 0083     14    CONTINUE
            C
ISN 0084           NM=NM+1
ISN 0085           A(NM,1)=1.DO
ISN 0086           A(NM,2)=10.DO*H
ISN 0087           A(NM,KK+3)=0.DO
            C
ISN 0088           NM1=NM+1
ISN 0089           NM2=NM+2
            C
ISN 0090           DO 11 I=NM1,NM2
ISN 0091              A(I,1)=0.DO
ISN 0092              A(I,2)=0.DO
ISN 0093              A(I,KK+3)=0.DO
ISN 0094     11    CONTINUE
            C
ISN 0095           DO 10 J=3,23
ISN 0096              A(NM1,J)=1.DO
ISN 0097              A(NM2,J)=DFLOAT(J-2)*H*10.DO
ISN 0098     10    CONTINUE
            C
```

```
ISN 0099              DO 17 J=24,KK2
ISN 0100                 A(NM1,J)=1.DO
ISN 0101                 A(NM2,J)=DFLOAT(210+(J-2-21))*H
ISN 0102        17    CONTINUE
                C
                C
ISN 0103              WRITE (6,63)
ISN 0104              WRITE (6,59) NM
ISN 0105              WRITE (6,63)
ISN 0106              DO 12 I=1,NM2
ISN 0107              WRITE (6,62) (A(I,J),J=1,11)
ISN 0108        12    CONTINUE
ISN 0109              WRITE (6,63)
ISN 0110              DO 13 I=1,NM2
ISN 0111              WRITE (6,62) (A(I,J),J=12,22)
ISN 0112        13    CONTINUE
ISN 0113              WRITE (6,63)
ISN 0114              DO 18 I=1,NM2
ISN 0115              WRITE (6,62) (A(I,J),J=23,33)
ISN 0116        18    CONTINUE
ISN 0117        62    FORMAT ((11(2X,E10.2)))
ISN 0118        63    FORMAT (///)
                C
                C
ISN 0119              IA=70
ISN 0120              NI=KK2
ISN 0121              L=1
ISN 0122              CALL DMAL2 (A,IA,NM2,NI,L)
                C
ISN 0123              DO 80 I=1,KK2
ISN 0124                 B(I)=A(I,KK+3)
ISN 0125        80    CONTINUE
                C
                C
ISN 0126        59    FORMAT (5X,'LE NOMBRE DE POINTS SUR LEQUEL SE FAIT LA DECONVOLUTIO
                     &N EST :',I6)
ISN 0127              RETURN
ISN 0128              END
```

/ DECONV / SIZE OF PROGRAM 013BB4 HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	SFA	R*8	00E610	B	S XR	R*8	000000	C	F XR	R*8	000000	H	F	R*8	0001F8
I	SFA	I*4	0001A8	J	SFA	I*4	0001AC	K	F C	I*4	000000	L	SFA	I*4	0001B0
M	SFA	I*4	0001B4	N	F C	I*4	000010	X	SF	R*8	000200	HC	SF	R*8	000208
HQ	SF	R*8	000210	HS	SF	R*8	000218	IA	SFA	I*4	0001B8	IJ	SF	I*4	0001BC
IL	SFA	I*4	0001C0	I1	SFA	I*4	0001C4	KK	SF	I*4	0001C8	K1	SF	I*4	0001CC
K2	S	I*4	0001D0	NI	SFA	I*4	0001D4	NM	SF	I*4	0001D8	NN	C	I*4	NR
N1	S	I*4	0001DC	N2	SF	I*4	0001E0	RC	SF	R*8	000238	RQ	SF	R*8	000918
TC	SF	R*8	000FF8	TQ	SF	R*8	0010F0	X2	SF	R*8	000220	X3	SF	R*8	000228
ILM	SFA	I*4	0001E4	KK2	SF	I*4	0001E8	KR2	S	I*4	0001EC	NM1	SF	I*4	0001F0
NM2	SFA	I*4	0001F4	TRS	SF	R*8	0011E8	KDIV	C	I*4	NR	KINS	C	I*4	NR
SIGM	F XR	R*8	000000	DMAL2	SF XF		000000	KREEL	F C	I*4	000004	DECONV		R*8	000230
IBCOM#	F XF	I*4	000000												

***** COMMON INFORMATION *****

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

   ISN 0002            SUBROUTINE RESULT (B,U,V,X,H)
   ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
   ISN 0004            COMMON /KONST/ K,KREEL,KINS,KDIV,N,NN
   ISN 0005            DIMENSION B(1),V(1),U(1),X(1)
   ISN 0006            DIMENSION RT(31)
                C
   ISN 0007            KR2=KREEL+2
   ISN 0008            K1=K+1
   ISN 0009            KK=K+9
                C
                C
   ISN 0010            DO 10 I=1,N
   ISN 0011               I1=I-1
   ISN 0012               V(I)=B(1)+B(2)*X(I)
   ISN 0013               DO 20 L=1,K
   ISN 0014                  IL=MAX(I1-10*L,0)
   ISN 0015                  RT(L)=DFLOAT(IL)**3
   ISN 0016                  RT(L)=RT(L)*(H**3)
   ISN 0017                  RT(L)=RT(L)/6.D0
   ISN 0018                  V(I)=V(I)+B(2+L)*RT(L)
   ISN 0019         20     CONTINUE
   ISN 0020               DO 30 L=K1,KK
   ISN 0021                  IL=MAX(I1-210-(L-21),0)
   ISN 0022                  RT(L)=DFLOAT(IL)**3
   ISN 0023                  RT(L)=RT(L)*(H**3)
   ISN 0024                  RT(L)=RT(L)/6.D0
   ISN 0025                  V(I)=V(I)+B(2+L)*RT(L)
   ISN 0026         30     CONTINUE
   ISN 0027         10  CONTINUE
                C
                C
   ISN 0028            WRITE (6,60)
   ISN 0029            WRITE (6,61) (U(I),V(I),I=1,N)
   ISN 0030         60 FORMAT (///,20X,'FONCTION TEST',20X,'FONCTION RESULTAT',//)
   ISN 0031         61 FORMAT ((20X,E14.7,20X,E14.7))
                C
   ISN 0032            RETURN
   ISN 0033            END
```

/ RESULT / SIZE OF PROGRAM 00053C HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
B	F XR	R*8	000000	H	F	R*8	000118	I	SF	I*4	0000F8	K	F C	I*4	000000
L	SFA	I*4	0000FC	N	F C	I*4	000010	U	F XR	R*8	000000	V	SF XR	R*8	000000
X	F XR	R*8	000000	IL	SFA	I*4	000100	I1	SFA	I*4	000104	KK	SF	I*4	000108
K1	SF	I*4	00010C	NN	C	I*4	NR	RT	SF	R*8	000128	KR2	S	I*4	000110
KDIV	C	I*4	NR	KINS	C	I*4	NR	KREEL	F C	I*4	000004	IBCOM#	F XF	I*4	000000
RESULT		R*8	000120												

***** COMMON INFORMATION *****

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

   ISN 0002            SUBROUTINE RECONV (B,C,S)
   ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
   ISN 0004            COMMON /KONST/ K,KREEL,KINS,KDIV,N,NN
   ISN 0005            DIMENSION B(1),C(1),S(1)
                  C
   ISN 0006            H=1.DO
   ISN 0007            N2=N-2
   ISN 0008            K1=K+1
   ISN 0009            KK=K+9
   ISN 0010            KK2=KK+2
                  C
   ISN 0011            HQ=H**4/24.DO
   ISN 0012            HC=H**5/120.DO
   ISN 0013            HS=H**7/5040.DO
                  C
   ISN 0014            DO 1 I=1,N
   ISN 0015               I1=I-1
   ISN 0016               X=I1*H
   ISN 0017               X2=X*X/2.DO
   ISN 0018               X3=X*X2/3.DO
                  C
   ISN 0019               Y=B(1)*(C(1)*X+C(2)*X2)
   ISN 0020               Z=B(2)*(C(1)*X2+C(2)*X3)
   ISN 0021               S(I)=Y+Z
   ISN 0022               DO 2 M=1,N2
   ISN 0023                  IM=MAX(I1-M,0)
   ISN 0024                  Y=(DFLOAT(IM)**4)*HQ*C(2+M)
   ISN 0025                  Z=(DFLOAT(IM)**5)*HC*C(2+M)
   ISN 0026                  S(I)=S(I)+B(1)*Y+B(2)*Z
   ISN 0027     2         CONTINUE
   ISN 0028               DO 3 L=1,K
   ISN 0029                  IL=MAX(I1-L*10,0)
   ISN 0030                  Y=(DFLOAT(IL)**4)*HQ*B(2+L)
   ISN 0031                  Z=(DFLOAT(IL)**5)*HC*B(2+L)
   ISN 0032                  S(I)=S(I)+C(1)*Y+C(2)*Z
   ISN 0033                  DO 4 M=1,N2
   ISN 0034                     ILM=MAX(I1-10*L-M,0)
   ISN 0035                     Z=(DFLOAT(ILM)**7)*HS*B(2+L)*C(2+M)
   ISN 0036                     S(I)=S(I)+Z
   ISN 0037     4               CONTINUE
   ISN 0038     3            CONTINUE
   ISN 0039               DO 5 L=K1,KK
   ISN 0040                  IL=MAX(I1-210-(L-21),0)
   ISN 0041                  Y=(DFLOAT(IL)**4)*HQ*B(2+L)
   ISN 0042                  Z=(DFLOAT(IL)**5)*HC*B(2+L)
   ISN 0043                  S(I)=S(I)+C(1)*Y+C(2)*Z
   ISN 0044                  DO 6 M=1,N2
   ISN 0045                     ILM=MAX(I1-210-(L-21)-M,0)
   ISN 0046                     Z=(DFLOAT(ILM)**7)*HS*B(2+L)*C(2+M)
   ISN 0047                     S(I)=S(I)+Z
   ISN 0048     6               CONTINUE
   ISN 0049     5            CONTINUE
   ISN 0050     1         CONTINUE
```

```
ISN 0051            WRITE (6,63)
ISN 0052      63    FORMAT (///)
ISN 0053            WRITE (6,60) ( S(I),C(I),B(I),I=1,N)
ISN 0054      60    FORMAT ((3(20X,E14.7)))
ISN 0055            WRITE (6,63)
             C
ISN 0056            RETURN
ISN 0057            END
```

/ RECONV / SIZE OF PROGRAM 00077C HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
B	F XR	R*8	000000	C	F XR	R*8	000000	H	SF	R*8	000130	I	SF	I*4	000100
K	F C	I*4	000000	L	SFA	I*4	000104	M	SFA	I*4	000108	N	F C	I*4	000010
S	SF XR	R*8	000000	X	SF	R*8	000138	Y	SF	R*8	000140	Z	SF	R*8	000148
HC	SF	R*8	000150	HQ	SF	R*8	000158	HS	SF	R*8	000160	IL	SFA	I*4	00010C
IM	SFA	I*4	000110	I1	SFA	I*4	000114	KK	SF	I*4	000118	K1	SF	I*4	00011C
NN	C	I*4	NR	N2	SF	I*4	000120	X2	SF	R*8	000168	X3	SF	R*8	000170
ILM	SFA	I*4	000124	KK2	S	I*4	000128	KDIV	C	I*4	NR	KINS	C	I*4	NR
KREEL	C	I*4	NR	IBCOM#	F XF	I*4	000000	RECONV		R*8	000178				

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK * KONST* SIZE OF BLOCK 000018 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
K	I*4	000000	KREEL	I*4	000004 NR	KINS	I*4	000008 NR	KDIV	I*4	00000C NR
N	I*4	000010	NN	I*4	000014 NR						

SOURCE STATEMENT LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
2	27	0003D0	4	37	0004CE	3	38	0004D6	6	48	000614
5	49	00061C	1	50	00064A						

COMPILER GENERATED LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
100001	2	000248	100002	15	000306	100003	23	00037A	100004	28	0003D8
100005	29	000410	100006	34	000488	100008	39	00050C	100009	40	000554
100010	45	0005CE	100013	51	000672	100014	53	0006D0	100015	53	0006EC

FORMAT STATEMENT LABELS '

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
63	52	000028	60	54	00002D						

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

STATISTICS SOURCE STATEMENTS = 56, PROGRAM SIZE = 1916, SUBPROGRAM NAME =RECONV

STATISTICS NO DIAGNOSTICS GENERATED

```
 LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED          DATE 82.281/10.14.11       PAGE    1

REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

   ISN 0002              SUBROUTINE COURBE (F,G,X,N,TGR)
                     C
   ISN 0003              DOUBLE PRECISION F(250),G(250),X(250),TGR
   ISN 0004              DIMENSION SF(250),SG(250),SX(250)
                     C
   ISN 0005              DO 1 I=1,N
   ISN 0006                 SF(I)=F(I)
   ISN 0007                 SG(I)=G(I)
   ISN 0008                 SX(I)=X(I)
   ISN 0009          1   CONTINUE
                     C
   ISN 0010              CALL TBRAF (1,20.6,20.,1,1HX,1,1HY,1,8,TGR,
                        1            SX(1),SF(1),-N,1H*,0 1,
                        2            SX(1),SG(1),+N,1H ,0 )
                     C
   ISN 0011              RETURN
   ISN 0012              END


                                                          /  COURBE /     SIZE OF PROGRAM 000E20 HEXADECIMAL BYTES

      NAME     TAG   TYPE  ADD.        NAME     TAG   TYPE  ADD.        NAME     TAG   TYPE  ADD.        NAME     TAG   TYPE  ADD.
        F   F  XR   R*8   000000          G   F  XR   R*8   000000          I SF        I*4   000108          N SFA       I*4   00010C
        X   F  XR   R*8   000000         SF SFA        R*4   000120         SG SFA      R*4   000508         SX SFA       R*4   0008F0
      TGR SFA       R*8   000118      TBRAF SF  XF              000000    COURBE        R*4   000110


 SOURCE STATEMENT LABELS

    LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR
        1    9  000D50

 COMPILER GENERATED LABELS

    LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR
   100001    2  000D14       100002    6  000D38       100003   10  000D56

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       11, PROGRAM SIZE =     3616, SUBPROGRAM NAME =COURBE

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                        186K BYTES OF CORE NOT USED

*STATISTICS*  NO  DIAGNOSTICS THIS STEP
```

ANNEXE V

LISTING D'AUTODÉCONVOLUTION

REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```
                C
ISN 0002              IMPLICIT REAL*8(A-H,O-Z)
                C
ISN 0003              EXTERNAL FONJAC
                C
ISN 0004              COMMON /DIVISE/ KDIV
ISN 0005              COMMON /CONST/ N,K,K2,MK,MK1,MK2,NEXP
ISN 0006              COMMON /PAS/ H,HPAS
ISN 0007              COMMON /SPLINE/ A(43,21),V(20),B(43)
ISN 0008              COMMON /INTERM/ C(43,41),D(43,41),E(43,41,41),T(43,86)
ISN 0009              COMMON /ABSC/ X(250)
ISN 0010              COMMON /POIDS/ P(43)
ISN 0011              COMMON /FONCT/ F(250),W(250),U(250),Z(250),FBR(250)
ISN 0012              COMMON /STAT/ FSUP,ALFA
ISN 0013              COMMON /COMPTE/ ITER
ISN 0014              DIMENSION SIGM (250),SIGM1(250)
ISN 0015              DIMENSION VP(250)
                C
                C
                C  N=NOMBRE DE POINTS
                C  RO=PARAMETRE DE LISSAGE
                C  H=PAS
                C  X=VECTEUR DES ABSCISSES
                C  K=NOMBRE DE SUBDIVISIONS DU SUPPORT DE LA SPLINE Z
                C  K2=K+2=NOMBRE DE PARAMETRES DE LA SPLINE Z
                C  U=VECTEUR CONTENANT LES VALEURS DE LA FONCTION TEST
                C  F=VECTEUR DES DONNEES
                C        F=U*U
                C  FBR=VECTEUR CONTENANT LES DONNEES BRUITEES
                C  SIGM=VECTEUR CONTENANT LES VALEURS DE LA SPLINE AUX NOEUDS
                C  SIGM1=VECTEUR CONTENANT LES VALEURS DE LA DERIVEE PREMIERE DE LA
                C    SPLINE AUX NOEUDS
                C  Z=VECTEUR CONTENANT LES VALEURS DE LA SPLINE DECONVOLEE
                C  W=Z*Z CALCULEE PAR LA METHODE DES TRAPEZES
                C  VP=VECTEUR DES N-2 VALEURS PROPRES DE LA MATRICE OMEGA
                C
                C
                C*******************************************************************
                C  APPEL DES DONNEES INITIALES
                C*******************************************************************
                C
ISN 0016              DATA TITR1 /8HRESULTAT/
ISN 0017              DATA TITR2 /8HLISSAGE /
ISN 0018              DATA TITR3 /8HANTEPOST/
                C
ISN 0019              CALL INIDO
                C
ISN 0020              HPAS=H*DFLOAT(KDIV)
                C
ISN 0021              DO 500 I=1,250
ISN 0022                 X(I)=DFLOAT(I-1)*H
ISN 0023        500   CONTINUE
                C
```

```
ISN 0024          CALL COEFF
                  C
                  C*****************************************************************
                  C  PURGE DE LA MEMOIRE
                  C*****************************************************************
                  C
ISN 0025          CALL PURGE (NEXP)
                  C
                  C*****************************************************************
                  C  BRUITAGE DE LA COURBE 'EXPERIMENTALE' F
                  C*****************************************************************
                  C
ISN 0026          IRAND=575757
ISN 0027          CALL BRUIT (NEXP,IRAND,F,FBR)
                  C
                  C*****************************************************************
                  C  LISSAGE DE LA COURBE ('EXPERIMENTALE') BRUITEE
                  C*****************************************************************
                  C
ISN 0028          ALFA=5.DO
ISN 0029          CALL SMOOTH (FBR,SIGM,SIGM1,VP,NEXP,H,RO)
                  C
                  C
ISN 0030          CALL COURBE (F,SIGM,X,NEXP.TITR3)
ISN 0031          CALL COURBE (FBR,SIGM,X,NEXP,TITR2)
                  C
                  C*****************************************************************
                  C   MODIFICATIONS DES SUPPORTS
                  C*****************************************************************
ISN 0032          DO 210 I=1,NEXP
ISN 0033             J=N-I+1
ISN 0034             SIGM(J)=SIGM(J-10)
ISN 0035             FBR(J)=FBR(J-10)
ISN 0036             F(J)=F(J-10)
ISN 0037      210 CONTINUE
                  C
ISN 0038          DO 220 I=1,10
ISN 0039             SIGM(I)=0.DO
ISN 0040             FBR(I)=0.DO
ISN 0041             F(I)=0.DO
ISN 0042      220 CONTINUE
                  C
                  C*****************************************************************
                  C   ON ENLEVE LES VALEURS NEGATIVES PROVENANT DU LISSAGE
                  C*****************************************************************
ISN 0043          J=0
ISN 0044          DO 200 I=1,N,KDIV
ISN 0045             J=J+1
ISN 0046             SIGM(J)=SIGM(I)
ISN 0047             FBR(J)=FBR(I)
ISN 0048             F(J)=F(I)
ISN 0049          IF (SIGM(J).GT.0.DO) GO TO 200
ISN 0051             SIGM(J)=0.DO
ISN 0052      200 CONTINUE
                  C
                  C*****************************************************************
                  C  APPROXIMATION DU SPECTRE DEBRUITE PAR UNE SPLINE AUTOCONVOLEE
                  C*****************************************************************
```

```
                C
ISN 0053              EPS=1.D-02
                C
ISN 0054              ITER=0
                C
ISN 0055              CALL DMAGT1 (A,K2,V,SIGM,K2,MK2,EPS,FONJAC,P)
                C
                C*********************************************************************
                C  EXPRESSION DE LA DECONVOLEE Z
                C*********************************************************************
                C
ISN 0056              DO 100 I=1,K2
ISN 0057                 B(I)=0.D0
ISN 0058                 DO 101 J= 2,MK1
ISN 0059                    B(I)=B(I)+T(I,K2+J)*V(J-1)
ISN 0060        101      CONTINUE
ISN 0061        100   CONTINUE
                C
ISN 0062              DO 102 I=1,N
ISN 0063                 I1=I-1
ISN 0064                 Z(I)=B(1)+B(2)*DFLOAT(I1)*H
ISN 0065                 DO 103 L=1,K
ISN 0066                    IL=MAX(I1-KDIV*L,0)
ISN 0067                    RT=DFLOAT(IL)**3
ISN 0068                    RT=RT*(H**3)
ISN 0069                    RT=RT/6.D0
ISN 0070                    Z(I)=Z(I)+B(2+L)*RT
ISN 0071        103      CONTINUE
ISN 0072        102   CONTINUE
                C
                C*********************************************************************
                C  CALCUL DE LA CONVOLEE DE Z PAR LA METHODE DES TRAPEZES
                C*********************************************************************
                C
ISN 0073              W(1)=0.D0
ISN 0074              N1=N-1
                C
ISN 0075              DO 110 I=1,N1
ISN 0076                    W(I+1)=0.D0
                C
ISN 0077                    DO 111 L=1,I
ISN 0078                          Z1=Z(L)/3.D0+Z(L+1)/6.D0
ISN 0079                          Z2=Z(L)/6.D0+Z(L+1)/3.D0
ISN 0080                          ZZ=Z(I-L+2)*Z1+Z(I-L+1)*Z2
ISN 0081                          W(I+1)=W(I+1)+H*ZZ
ISN 0082        111               CONTINUE
                C
ISN 0083        110   CONTINUE
                C
                C*********************************************************************
                C  CALCUL D'ERREURS
                C*********************************************************************
                C
ISN 0084              EPSIS=0.
ISN 0085              EPSIW=0.
ISN 0086              EPSIF=0.
ISN 0087              EPSIRE=0.
                C
```

```
                C-----ERREURS ABSOLUES-----
                C
ISN 0088              DO 120 I=1,N
ISN 0089                    EPSIZ=EPSIZ+(U(I)-Z(I))**2
ISN 0090                    EPSIW=EPSIW+(FBR(I)-W(I))**2
ISN 0091                    EPSIF=EPSIF+(F(I)-W(I))**2
ISN 0092        120   CONTINUE
ISN 0093        140   CONTINUE
                C
                C-----MOYENNE DES ERREURS-----
                C
ISN 0094              EPSIZ=EPSIZ/DFLOAT(N)
ISN 0095              EPSIW=EPSIW/DFLOAT(N)
ISN 0096              EPSIF=EPSIF/DFLOAT(N)
                C
                C
                C*************************************************************************
                C  APPEL DES TRACES DE COURBES
                C*************************************************************************
                C
ISN 0097              CALL COURBE (U,Z,X,N,TITR1)
                C
                C
                C
ISN 0098              J=0
ISN 0099              DO 230 I=1,N,KDIV
ISN 0100                 J=J+1
ISN 0101                 W(J)=W(I)
ISN 0102        230   CONTINUE
                C
                C
                C*************************************************************************
                C  ORDRES D'IMPRESSION
                C*************************************************************************
                C
ISN 0103              WRITE (6,611)
ISN 0104              WRITE (6,604)
ISN 0105              WRITE (6,604)
ISN 0106              WRITE (6,604)
ISN 0107              WRITE (6,604)
ISN 0108              WRITE (6,604)
ISN 0109              WRITE (6,600)
ISN 0110              WRITE (6,601) N
ISN 0111              WRITE (6,602) H
ISN 0112              WRITE (6,603) K2
ISN 0113              WRITE (6,613) RO
ISN 0114              WRITE (6,604)
ISN 0115              WRITE (6,621) EPS
ISN 0116              WRITE (6,615) ALEA
ISN 0117              WRITE (6,605) ITER
ISN 0118              WRITE (6,607) EPSIZ
ISN 0119              WRITE (6,606) EPSIW
ISN 0120              WRITE (6,614) EPSIF
ISN 0121              WRITE (6,611)
ISN 0122              WRITE (6,604)
ISN 0123              WRITE (6,604)
ISN 0124              WRITE (6,608)
ISN 0125              WRITE (6,609)
```

```
ISN 0126            WRITE (6,612)
ISN 0127            DO 130 I=1,K2
ISN 0128               WRITE (6,610) (F(I),FBR(I),SIGM(I),A(I,MK),W(I))
ISN 0129      130   CONTINUE
ISN 0130            WRITE (6,611)
ISN 0131            WRITE (6,604)
ISN 0132            WRITE (6,604)
ISN 0133            WRITE (6,623)
ISN 0134            WRITE (6,624)
ISN 0135            DO 150 I=1,N
ISN 0136               WRITE (6,622) (U(I),Z(I))
ISN 0137      150   CONTINUE
ISN 0138            WRITE (6,611)
ISN 0139            WRITE (6,604)
ISN 0140            WRITE (6,604)
ISN 0141            WRITE (6,604)
                C
                C*************************************************************************
                C  FORMATS D'IMPRESSION
                C*************************************************************************
                C
ISN 0142      600    FORMAT (40X,51(1H*),/,40X,51H* DECONVOLUTION PAR FONCTIONS SPLINE
                     *D'AJUSTEMENT *,/,40X,51(1H*),///)
ISN 0143      601    FORMAT (2X,'IL Y A ',I3,' POINTS EXPERIMENTAUX CONSIDERES',/)
ISN 0144      603    FORMAT (2X,'LE NOMBRE DE PARAMETRES EST ',I3,/)
ISN 0145      602    FORMAT (2X,'LE PAS DE LA SUBDIVISION EST DE ',F5.3,/)
ISN 0146      604    FORMAT (//)
ISN 0147      605    FORMAT (20X,'ITER=',I4,/)
ISN 0148      606    FORMAT (20X,'EPSIW=',E19.12,/)
ISN 0149      607    FORMAT (20X,'EPSIZ=',E19.12,/)
ISN 0150      608    FORMAT (10X,10HFONCTION F,13X,20HFONCTION BRUITEE FBR,3X,13HFONCTI
                     &ON SIGM,10X,17HFONCTION SPLINE A,6X,12HRECONVOLEE W)
ISN 0151      609    FORMAT (10X,15H'EXPERIMENTALE',8X,20(1H*),3X,10HDE LISSAGE,13X,20H
                     &AUTOCONVOLEE AJUSTEE,3X,12(1H*))
ISN 0152      610    FORMAT (10X,4(E13.6,10X),E13.6)
ISN 0153      611    FORMAT (1H1)
ISN 0154      612    FORMAT (10X,15(1H*),31X,13(1H*),10X,20(1H*),//)
ISN 0155      613    FORMAT (2X,45HLA VALEUR DU PARAMETRE D'AJUSTEMENT EST RO = ,E8.2,/
                     @)
ISN 0156      614    FORMAT (20X,'EPSIF=',E19.12,/)
ISN 0157      615    FORMAT (20X,'ALEA=',E8.2,/)
ISN 0158      617    FORMAT (40X,'PARAMETRES DE SORTIE')
ISN 0159      618    FORMAT (40X,20(1H*),//)
ISN 0160      619    FORMAT (43X,E19.12)
ISN 0161      621    FORMAT (20X,'EPSILON = ',E8.1,/)
ISN 0162      622    FORMAT (36X,E13.6,28X,E13.6)
ISN 0163      623    FORMAT (35X,15HFONCTION TEST U,20X,28HFONCTION SPLINE DECONVOLEE Z
                     &)
ISN 0164      624    FORMAT (35X,15(1H*),20X,28(1H*),//)
                C
                C
ISN 0165            STOP
ISN 0166            END
```

/ MAIN / SIZE OF PROGRAM 002664 HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

   ISN 0002              SUBROUTINE FONJAC
                C
   ISN 0003              IMPLICIT REAL*8(A-H,O-Z)
                C
   ISN 0004              COMMON /DIVISE/ KDIV
   ISN 0005              COMMON /CONST/ N,K,K2,MK,MK1,MK2,NEXP
   ISN 0006              COMMON /SPLINE/ A(43,21),V(20),B(43)
   ISN 0007              COMMON /INTERM/ C(43,41),D(43,41),E(43,41,41),T(43,86)
   ISN 0008              COMMON /ABSC/ X(250)
   ISN 0009              COMMON /FONCT/ F(250),W(250),U(250),Z(250),FBR(250)
   ISN 0010              COMMON /COMPTE/ ITER
   ISN 0011              DIMENSION AZ(43),Q(43,43)
   ISN 0012              DIMENSION Y(43)
                C
                C  COMPTE DU NOMBRE D'ITERATIONS
                C  ==============================
                C
   ISN 0013              ITER=ITER+1
   ISN 0014              WRITE (6,60) ITER
   ISN 0015         60   FORMAT (2X,'ITER=',I4)
                C
                C*********************************************************************
                C  ENTREE DES CONTRAINTES LINEAIRES
                C*********************************************************************
                C
   ISN 0016              DO 110 I=1,K2
   ISN 0017                 B(I)=0.D0
   ISN 0018                 DO 120 J=2,MK1
   ISN 0019                    B(I)=B(I)+T(I,K2+J)*V(J-1)
   ISN 0020         120     CONTINUE
   ISN 0021         110  CONTINUE
                C
                C*********************************************************************
                C  CALCUL DES ABSCISSES D'INTEGRATION
                C*********************************************************************
                C
   ISN 0022              J=0
   ISN 0023              DO 50 I=1,N,KDIV
   ISN 0024                 J=J+1
   ISN 0025                 Y(J)=X(I)
   ISN 0026         50   CONTINUE
                C
                C
   ISN 0027              DO 10 I=1,K2
                C
                C*********************************************************************
                C  CALCUL DU JACOBIEN
                C*********************************************************************
                C
                C  NOTATIONS
                C  ---------
   ISN 0028              X1=Y(I)
   ISN 0029              X2=(X1*X1)/2.
```

```
ISN 0030            X3=(X2*X1)/3.
                C
                C  CALCUL DES DERIVEES PARTIELLES PAR RAPPORT A B(1) ET B(2)
                C  ________________________________________________________
                C
ISN 0031            Q(I,1)=(B(1)*X1+B(2)*X2)
ISN 0032            Q(I,2)=(B(1)*X2+B(2)*X3)
ISN 0033               DO 11 L=1,K
ISN 0034               Q(I,1)=Q(I,1)+C(I,L)*B(2+L)
ISN 0035               Q(I,2)=Q(I,2)+D(I,L)*B(2+L)
ISN 0036         11    CONTINUE
ISN 0037            Q(I,1)=2.*Q(I,1)
ISN 0038            Q(I,2)=2.*Q(I,2)
                C
                C  CALCUL DES AUTRES DERIVEES PARTIELLES
                C  _____________________________________
                C
ISN 0039               DO 12 J=1,K
ISN 0040               Q(I,J+2)=B(1)*C(I,J)+B(2)*D(I,J)
ISN 0041               DO 13 L=1,K
ISN 0042               Q(I,J+2)=Q(I,J+2)+E(I,J,L)*B(2+L)
ISN 0043         13    CONTINUE
ISN 0044               Q(I,J+2)=2.DO*Q(I,J+2)
ISN 0045         12    CONTINUE
                C
                C  PRODUIT DES DERIVEES PARTIELLES PAR LES APPLICATIONS LINEAIRES
                C  ______________________________________________________________
                C
ISN 0046               DO 20 J=1,MK1
ISN 0047               AJ=0.DO
ISN 0048               DO 21 L=1,K2
ISN 0049               AJ=AJ+Q(I,L)*T(L,K2+J)
ISN 0050         21    CONTINUE
ISN 0051               AZ(J)=AJ
ISN 0052         20    CONTINUE
ISN 0053               DO 22 J=1,MK2
ISN 0054               A(I,J)=AZ(J+1)
ISN 0055         22    CONTINUE
                C
                C*********************************************************************
                C  CALCUL DE LA FONCTION
                C*********************************************************************
                C
ISN 0056               A(I,MK)=B(1)*(B(1)*X1+B(2)*X2)+B(2)*(B(1)*X2+B(2)*X3)
ISN 0057               DO 16 L=1,K
ISN 0058               A(I,MK)=A(I,MK)+2.DO*B(2+L)*(B(1)*C(I,L)+B(2)*D(I,L))
ISN 0059               DO 17 J=1,K
ISN 0060               A(I,MK)=A(I,MK)+E(I,L,J)*B(2+L)*B(2+J)
ISN 0061         17    CONTINUE
ISN 0062         16    CONTINUE
                C
                C
ISN 0063         10  CONTINUE
                C
                C
ISN 0064            RETURN
ISN 0065            END
```

REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```
ISN 0002            SUBROUTINE INIDO
                C
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
                C
ISN 0004            COMMON /DIVISE/ KDIV
ISN 0005            COMMON /PAS/ H,HPAS
ISN 0006            COMMON /CONST/ N,K,K2,MK,MK1,MK2,NEXP
ISN 0007            COMMON /SPLINE/ A(43,21),V(20),B(43)
ISN 0008            COMMON /FONCT/ F(250),W(250),U(250),Z(250),FBR(250)
ISN 0009            COMMON /ABSC/ X(250)
ISN 0010            COMMON /STAT/ FSUP,ALFA
ISN 0011            DIMENSION Y(250)
                C
                C*************************************************************************
                C  LECTURE DES DONNEES
                C*************************************************************************
                C
ISN 0012            PI=2.DO*DARSIN(1.DO)
ISN 0013            KDIV=5
ISN 0014            N=211
                C   K=((N+KDIV-1)/KDIV)-2
ISN 0015            K=41
ISN 0016            K2=K+2
ISN 0017            MK=21
ISN 0018            MK1=MK-1
ISN 0019            MK2=MK-2
                C
                C*************************************************************************
                C  CONSTANTES UTILES
                C*************************************************************************
                C
ISN 0020            NEXP=N-10
ISN 0021            N10=NEXP
ISN 0022            N5=N-5
                C
ISN 0023            IOTA=1.D-01
ISN 0024            BETA=5.D-01
ISN 0025            DELTA=3.D-01
ISN 0026            FSUP=0.DO
                C
ISN 0027            H=PI/40.DO
                C
ISN 0028            DO 70 I=1,250
ISN 0029               Y(I)=DFLOAT(I-1)*H
ISN 0030        70  CONTINUE
                C
                C
ISN 0031            DO 1 I=1,250
ISN 0032               U(I)=0.DO
ISN 0033               F(I)=0.DO
ISN 0034        1   CONTINUE
                C
                C  ENTREE DE LA FONCTION TEST (FONCTION TRIGONOMETRIQUE A TROIS BOSSES)
```

```
                C  ======================================================================
                C
ISN 0035              DO 310 I=1,81
ISN 0036                 U(I)=(1.DO-DCOS(Y(I)))*IOTA
ISN 0037        310    CONTINUE
ISN 0038              DO 320 I=61,141
ISN 0039                 U(I)=U(I)+(1.DO+DSIN(Y(I)))*BETA
ISN 0040        320    CONTINUE
ISN 0041              DO 330 I=121,201
ISN 0042                 U(I)=U(I)+(1.DO+DCOS(Y(I)))*DELTA
ISN 0043        330    CONTINUE
                C
ISN 0044              J=0
ISN 0045              DO 340 I=1,201,2
ISN 0046                 J=J+1
ISN 0047                 U(J)=U(I)
ISN 0048        340   CONTINUE
                C
ISN 0049              J=J+1
ISN 0050              DO 350 I=J,N
ISN 0051                 U(I)=0.DO
ISN 0052        350   CONTINUE
                C
                C
ISN 0053              H=1.DO
                C
                C
                C  CALCUL DE LA CONVOLEE PAR TRAPEZES : U*U
                C  =========================================
                C
ISN 0054              N1=N10-1
ISN 0055              DO 2 I=1,N1
ISN 0056                 DO 3 J=1,I
ISN 0057                    F1=U(J)/3.DO+U(J+1)/6.DO
ISN 0058                    F2=U(J)/6.DO+U(J+1)/3.DO
ISN 0059                    F3=U(I-J+2)*F1+U(I-J+1)*F2
ISN 0060                    F(I+1)=F(I+1)+H*F3
ISN 0061                    FSUP=DMAX1(FSUP,F(I+1))
ISN 0062        3          CONTINUE
ISN 0063        2       CONTINUE
                C
                C
                C  DECALAGE ARTIFICIEL DES DONNEES
                C
ISN 0064              DO 11 I=1,N5
ISN 0065                 J=N-I+1
ISN 0066                 U(J)=U(J-5)
ISN 0067        11     CONTINUE
                C
ISN 0068              DO 12 I=1,5
ISN 0069                 U(I)=0.DO
ISN 0070        12     CONTINUE
                C
                C
                C**********************************************************************
                C  INITIALISATION DES V(I)
                C**********************************************************************
ISN 0071              DO 5 I=1,MK2
```

```
ISN 0072              V(I)=20.DO
ISN 0073      5     CONTINUE
               C
ISN 0074            RETURN
ISN 0075            END
```

/ INIDO / SIZE OF PROGRAM 000D28 HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
A	C	R*8	NR	B	C	R*8	NR	F SFA	C	R*8	000000	H SF	C	R*8	000000
I SFA		I*4	000124	J SF		I*4	000128	K SF	C	I*4	000004	N SF	C	I*4	000000
U SF	C	R*8	000FA0	V S	C	R*8	001C38	W	C	R*8	NR	X	C	R*8	NR
Y SFA		R*8	000170	Z	C	R*8	NR	F1 SF		R*8	000140	F2 SF		R*8	000148
F3 SF		R*8	000150	K2 S	C	I*4	000008	MK SF	C	I*4	00000C	N1 SF		I*4	00012C
N5 SF		I*4	000130	PI SF		R*8	000158	FBR	C	R*8	NR	MK1 S	C	I*4	000010
MK2 SF	C	I*4	000014	N10 SF		I*4	000134	ALFA	C	R*8	NR	BETA SF		R*8	000160
DCOS F	XF	R*8	000000	DSIN F	XF	R*8	000000	FSUP SFA	C	R*8	000000	HPAS	C	R*8	NR
IOTA SF		I*4	000138	KDIV S	C	I*4	000000	NEXP SF	C	I*4	000018	DELTA SF		R*8	000168
INIDO		I*4	00013C	DARSIN F	XF	R*8	000000								

***** COMMON INFORMATION *****

NAME OF COMMON BLOCK *DIVISE* SIZE OF BLOCK 000004 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL	ADDR
KDIV	I*4	000000										

NAME OF COMMON BLOCK * PAS* SIZE OF BLOCK 000010 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL	ADDR.	VAR	NAME	TYPE	REL	ADDR	VAR	NAME	TYPE	REL	ADDR
H	R*8	000000	HPAS	R*8	000008	NR										

NAME OF COMMON BLOCK * CONST* SIZE OF BLOCK 00001C HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.	VAR. NAME	TYPE	REL. ADDR.
N	I*4	000000	K	I*4	000004	K2	I*4	000008	MK	I*4	00000C
MK1	I*4	000010	MK2	I*4	000014	NEXP	I*4	000018			

NAME OF COMMON BLOCK *SPLINE* SIZE OF BLOCK 001E30 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL. ADDR	VAR	NAME	TYPE	REL	ADDR	VAR	NAME	TYPE	REL	ADDR	VAR	NAME	TYPE	REL	ADDR
A	R*8	000000	NR	V	R*8	001C38			B	R*8	001CD8	NR					

NAME OF COMMON BLOCK * FONCT* SIZE OF BLOCK 002710 HEXADECIMAL BYTES

VAR. NAME	TYPE	REL	ADDR	VAR	NAME	TYPE	REL	ADDR	VAR	NAME	TYPE	REL	ADDR	VAR. NAME	TYPE	REL. ADDR.
F	R*8	000000			W	R*8	0007D0	NR		U	R*8	000FA0		Z	R*8	001770 NR
FBR	R*8	001F40	NR													

NAME OF COMMON BLOCK * ABSC* SIZE OF BLOCK 0007D0 HEXADECIMAL BYTES

REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

```
ISN 0002            SUBROUTINE COEFF
                C
ISN 0003            IMPLICIT REAL*8(A-H,O-Z)
                C
ISN 0004            COMMON /DIVISE/ KDIV
ISN 0005            COMMON /PAS/ H,HPAS
ISN 0006            COMMON /CONST/ N,K,K2,MK,MK1,MK2,NEXP
ISN 0007            COMMON /INTERM/ C(43,41),D(43,41),E(43,41,41),T(43,86)
ISN 0008            COMMON /POIDS/ P(43)
                C
                C*******************************************************************
                C  CALCUL DES COEFFICIENTS INTERVENANT DANS LE MODELE DE FONCTION SPLINE
                C  UTILISE
                C*******************************************************************
                C
ISN 0009            J=0
ISN 0010            K1=K+1
                C
ISN 0011            DO 300 I=1,N,KDIV
ISN 0012               J=J+1
ISN 0013               J2=J-2
ISN 0014               IF (J.LT.3) GO TO 300
ISN 0016               DO 301 L=1,J2
ISN 0017                  CD=DFLOAT(J-1-L)*HPAS
ISN 0018                  C(J,L)=CD**4/24.DO
ISN 0019                  D(J,L)=CD**5/120.DO
ISN 0020                  JL2=J2-L
ISN 0021                  IF (JL2.LT.1) GO TO 300
ISN 0023                  DO 302 M=1,JL2
ISN 0024                     CDE=DFLOAT(J-1-L-M)*HPAS
ISN 0025                     E(J,L,M)=CDE**7/5040.DO
ISN 0026        302          CONTINUE
ISN 0027        301       CONTINUE
ISN 0028        300    CONTINUE
                C
                C  INITIALISATION DE LA MATRICE T
                C  ==============================
                C
                C     AU DEPART T EST LA MATRICE TELLE QUE :
                C                                        V=TB
                C
ISN 0029            DO 1000 I=1,41
ISN 0030               DO 1001 J=1,82
ISN 0031                  T(I,J)=0.DO
ISN 0032        1001      CONTINUE
ISN 0033        1000   CONTINUE
                C
ISN 0034            DO 100 I=1,K
ISN 0035               T(I,1)=1.DO
ISN 0036               T(I,2)=DFLOAT(I)*HPAS
ISN 0037               IF (I.LT.2) GO TO 100
ISN 0039               I1=I+1
ISN 0040               DO 101 J=3,I1
```

```
ISN 0041                Z=DFLOAT(I+2-J)*HPAS
ISN 0042                T(I,J)=Z**3/6.DO
ISN 0043      101     CONTINUE
ISN 0044      100   CONTINUE
ISN 0045            DO 110 J=3,K2
ISN 0046               T(K1,J)=1.DO
ISN 0047               T(K2,J)=DFLOAT(J-2)*HPAS
ISN 0048      110   CONTINUE
              C
              C
              C  TRIANGULATION DE LA MATRICE T
              C  ==============================
              C
ISN 0049            DO 120 I=1,K2
ISN 0050               T(I,K2+I)=1.DO
ISN 0051      120   CONTINUE
              C
ISN 0052            DO 200 L=2,K2
ISN 0053               I=K2-L+1
ISN 0054               I1=I+1
ISN 0055               ALFA=T(I,I1)/T(I1,I1)
ISN 0056               DO 201 J=1,K2
ISN 0057                  T(I,K2+J)=T(I,K2+J)-T(I1,K2+J)*ALFA
ISN 0058      201      CONTINUE
ISN 0059               DO 202 J=1,I1
ISN 0060                  T(I,J)=T(I,J)-T(I1,J)*ALFA
ISN 0061      202      CONTINUE
ISN 0062               T(I,I1)=0.DO
ISN 0063      200   CONTINUE
              C
              C  INVERSION DE T
              C  ==============
              C     A PARTIR DE MAINTENANT T EST TELLE QUE:
ISN 0064                                                  B=TV
              C
ISN 0065            DO 400 L=1,K2
ISN 0066               T(1,K2+L)=T(1,K2+L)/T(1,1)
ISN 0067               DO 401 I=2,K2
ISN 0068                  I1=I-1
ISN 0069                  DO 402 J=1,I1
ISN 0070                     T(I,K2+L)=T(I,K2+L)-T(I,J)*T(J,K2+L)
ISN 0071      402         CONTINUE
ISN 0072                  T(I,K2+L)=T(I,K2+L)/T(I,I)
ISN 0073      401      CONTINUE
ISN 0074      400   CONTINUE
              C
              C*******************************************************************
              C  CALCUL DE LA FONCTION DE POIDS
              C*******************************************************************
              C
ISN 0075            DO 30 I=1,K2
ISN 0076               P(I)=1.DO
ISN 0077      30    CONTINUE
              C
ISN 0078            RETURN
ISN 0079            END
```

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

   ISN 0002            SUBROUTINE PURGE (N)
                  C
                  C
   ISN 0003            IRAND=575757
   ISN 0004            VMOYEN=0.
   ISN 0005            VSIGMA=1.
   ISN 0006            DO 1 I=1,N
   ISN 0007               A=APHERF(IRAND,VMOYEN,VSIGMA)
   ISN 0008       1    CONTINUE
   ISN 0009            RETURN
   ISN 0010            END

                                                    /    PURGE /     SIZE OF PROGRAM 00011A HEXADECIMAL BYTES

      NAME    TAG   TYPE  ADD.        NAME    TAG   TYPE  ADD.        NAME    TAG   TYPE  ADD.        NAME    TAG   TYPE  ADD.
         A S        R*4   000098          I SF        I*4   00009C          N  F        I*4   0000AO      IRAND SFA       I*4   0000A4
     PURGE          R*4   0000A8     APHERF  F  XF   R*4   000000     VMOYEN SFA       R*4   0000AC     VSIGMA SFA       R*4   0000BO


  SOURCE STATEMENT LABELS

    LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR
        1    8  0000EC

  COMPILER GENERATED LABELS

    LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR
   100001    2  0000BC       100002    7  0000DE       100003    9  0000FO

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       9, PROGRAM SIZE =      282, SUBPROGRAM NAME = PURGE

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                  186K BYTES OF CORE NOT USED
```

```
 LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED          DATE 82.281/10.10.17       PAGE    1

REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

   ISN 0002              SUBROUTINE BRUIT (N,IRAND,F,FBR)
                C
   ISN 0003              DOUBLE PRECISION F(1),FBR(1)
   ISN 0004              DOUBLE PRECISION H
                C
                C
   ISN 0005              DO 1 I=1,N
   ISN 0006                    VMOYEN=F(I)
   ISN 0007                    VMOYEN=ABS(VMOYEN)
   ISN 0008                    VSIGMA=SQRT(VMOYEN)
   ISN 0009                    FBR(I)=APHERF(IRAND,VMOYEN,VSIGMA)
   ISN 0010         1    CONTINUE
                C
                C
   ISN 0011              RETURN
   ISN 0012              END


                                                        /   BRUIT /     SIZE OF PROGRAM 0001A8 HEXADECIMAL BYTES

     NAME     TAG   TYPE  ADD.      NAME     TAG   TYPE  ADD      NAME     TAG   TYPE  ADD.      NAME     TAG   TYPE  ADD.
        F  F  XR    R*8   000000         H          R*8    NR          I  SF      I*4   000094         N  F      I*4   000098
      FBR  S  XR    R*8   000000      SQRT  F  XF   R*4   000000   BRUIT           R*4   00009C     IRAND  FA     I*4   0000A0
   APHERF  F  XF    R*4   000000    VMOYEN SFA      R*4   0000A4   VSIGMA SFA      R*4   0000A8


  SOURCE STATEMENT LABELS

    LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR
        1   10  000134

  COMPILER GENERATED LABELS

    LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR          LABEL  ISN   ADDR
   100001    2  0000E4       100002    6  0000FC       100003   11  000138

*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*      SOURCE STATEMENTS =       11, PROGRAM SIZE =       424, SUBPROGRAM NAME = BRUIT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                  186K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: EB,NODECK,NOFMT,GOSTMT,NOLIST,MAP,OPT(2),OBJ,NOALC,NOANSF,S,TERM,NOXREF

OPTIONS IN EFFECT: NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

   ISN 0002            SUBROUTINE COURBE (F,G,X,N,TGR)
                C
                C      NE PAS OUBLIER DE METTRE :  DATA TGR /8H......../
                C      DANS LE PROGRAMME PRINCIPAL
                C
   ISN 0003             DOUBLE PRECISION F(N),G(N),X(N),TGR
   ISN 0004             DIMENSION SF(250),SG(250),SX(250)
                C
                C  CONVERSION EN SIMPLE PRECISION
                C  ==============================
   ISN 0005             DO 1 I=1,N
   ISN 0006                SF(I)=F(I)
   ISN 0007                SG(I)=G(I)
   ISN 0008                SX(I)=X(I)
   ISN 0009        1    CONTINUE
                C
                C  APPEL DU SOUS-PROGRAMME DE TRACES DE COURBES
                C  ============================================
                C
   ISN 0010             CALL TBRAF (1,20.6,20.,1,1HX,1,1HY,1,8,TGR,
                       1            SX(1),SF(1),-N,1H*,0.1,
                       2            SX(1),SG(1),+N,1H.,0.)
                C
   ISN 0011             RETURN
   ISN 0012             END
```

/ COURBE / SIZE OF PROGRAM 000E20 HEXADECIMAL BYTES

NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.	NAME	TAG	TYPE	ADD.
F	F XR	R*8	000000	G	F XR	R*8	000000	I	SF	I*4	000108	N	SFA	I*4	00010C
X	F XR	R*8	000000	SF	SFA	R*4	000120	SG	SFA	R*4	000508	SX	SFA	R*4	0008F0
TGR	SFA	R*8	000118	TBRAF	SF XF		000000	COURBE		R*4	000110				

SOURCE STATEMENT LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
1	9	000D50									

COMPILER GENERATED LABELS

LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR	LABEL	ISN	ADDR
100001	2	000D14	100002	6	000D38	100003	10	000D56			

```
*OPTIONS IN EFFECT*NAME(MAIN) OPTIMIZE(2) LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT MAP NOFORMAT GOSTMT NOXREF NOALC NOANSF TERM IBM FLAG(I)

*STATISTICS*      SOURCE STATEMENTS =       11, PROGRAM SIZE =      3616, SUBPROGRAM NAME =COURBE

*STATISTICS*  NO  DIAGNOSTICS GENERATED
```